FUMETTO DEL MESE:
"TUTTA COLPA DEL VIRUS"

25 settembre 1988 - Anno VII - N. 56
Sped. Abb. Post. Gr. III/70 - Distr. MePe

Esclusivo per C64

Un programma per emulare i colori dell'Amiga

TUTTE LE SYS DEL C64

128 IN CP/M

C16 E PLUS 4, DA 40 AD 80 COLONNE

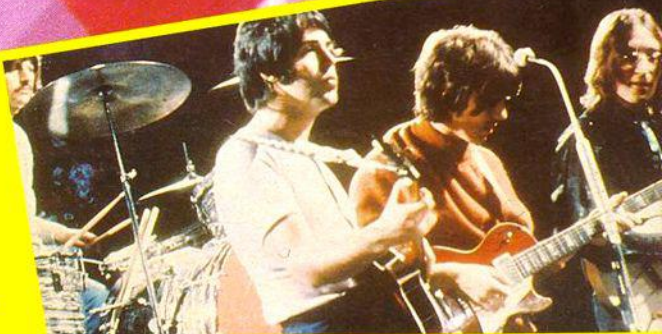
BIT PARADE

WIZBALL

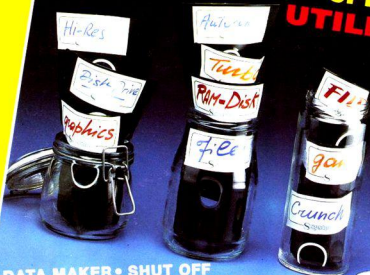
Sensazionale

Ora il C64 legge e suona le cassette di musica

IN EDICOLA

Commodore COMPUTER CLUB

L. 15.000

SPECIALE UTILITIES

Hi-Res · Disk Drive · graphics · Autorun · Turbo · RAM-Disk · File · Crunch

- DATA MAKER • SHUT OFF
- 4WD-RAMDISK • KOALA READER
- ASSEMBLER • DISK MERGER • TURBOKIT
- DISK MANAGER • AUTORUN MAKER • CRUNCHER
- TYPE • FORMAT SAVER • CROSS REFERENCE
- GRAPH COMPILER • RENUMBER • DELTAFIGHTER

in omaggio

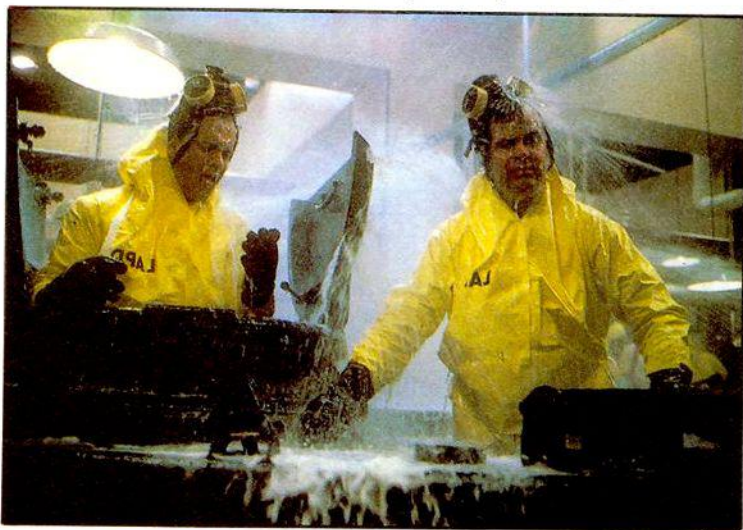
IL SUPERGAME DELTAFIGHTER

Sommario

RUBRICHE

Direttore: Alessandro de Simone - **Caporedattore:** Michele Maggi
Redazione/collaboratori: Paolo Agostini, Davide Ardizzone, Claudio Baiocchi, Luigi Callegari, Sergio Camici, Umberto Colapicchioni, Maurizio Dell'Abate, Valerio Ferri, Roberto Ferro, Cristina Magnaghi, Giancarlo Mariani, Roberto Marigo, Clizio Merli, Marco Miotta, Marco Miotti, Oscar Moccia, Roberto Morassi, Guido Pagani, Antonio Pastorelli, Sonja Scharrer, Fabio Sorgato, Valentino Spataro, Danilo Toma
Grafica: Arturo Ciaglia, Elena Salvadori
Direzione, redazione, pubblicità: v.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
Pubblicità: Milano: Leandro Nencioni (direttore vendite), Giorgio Ruffoni - Viale Famagosta, 75 - 20142 Milano Tel. 02/8467348
• Emilia Romagna: Spazio E - P.zza Roosevelt, 4 - 40123 Bologna - Tel. 051/236979
• Toscana, Marche, Umbria: Mercurio srl - via Rodari, 9 - San Giovanni Valdarno (Ar) - Tel. 055/947444
• Lazio, Campania: Spazio Nuovo - via P. Foscari, 70 - 00139 Roma - Tel. 06/8109679
Segreteria: Tiziana Sodano - **Abbonamenti:** Liliana Spina
Tariffe: prezzo per copia L. 4.500. Abbonamento annuo (11 fascicoli) L. 45.000. Estero: il doppio.
Abbonamento cumulativo alle riviste Computer e Commodore Computer Club L. 85.000.
I versamenti vanno indirizzati a: Systems Editoriale Srl mediante assegno bancario o utilizzando il c/c postale n. 37952207
Composizione: Systems Editoriale Srl - **Fotolito:** Systems Editoriale Srl
Stampa: Systems Editoriale/La Litografica Srl - Busto Arsizio (Va)
Registrazioni: Tribunale di Milano n. 370 del 2/10/82 - Direttore Responsabile: Michele Di Pisa
Sped. in abb. post. gr. III - Pubblicità inferiore al 70%
Distrib.: MePe - via G. Carcano, 32 - Milano
Periodici Systems: Banca Oggi - Commodore Club (disco) - Commodore Computer Club - Commodore Computer Club (disco produzione tedesca) - Computer - Computer disco - Electronic Mass Media Age - Energy Manager - Hospital Management - MondoRicambi - Nursing '90 - PC Programm (disco) - Personal Computer - Security - Software Club (cassetta ed. italiana) - VR Videoregistrare

GRAZIE, FARMOPLANT!

Un computer è davvero così "limitato" come vogliono farci credere? Proviamo ad immaginare un mondo governato esclusivamente da cervelli al silicio, potremmo imparar qualcosa...

Mentre scrivo queste note (seconda metà di luglio) le notizie relative al noto disastro ecologico pervengono sempre più aggiornate e preoccupanti: chilometri di litorale interdetti ai bagnanti, turisti in fuga, economia congelata, abitanti quantomeno perplessi di fronte alla disinvoltura con cui viene gestita la salute (e la finanza) pubblica.

E allora grazie, Farmoplant, per aver delimitato con maggiore precisione i limiti esistenti tra cervelli elettronici e cervelli umani.

Un qualsiasi C/64, infatti, operando con la propria indiscussa e limitata "intelligenza" avrebbe avvertito con un'insormontabile segnalazione di errore (del tipo, suppongo, "Illegal quantity") l'elaborazione relativa ai parametri di sicurezza usati nella fabbrica.

Allo stesso modo, grazie al lavoro di un perspicace ragazzino appassionato di informatica, un semplice spreadsheet, opportunamente programmato, si sarebbe rifiutato di accettare parametri incongruenti, avrebbe segnalato con allarmante precisione, nero su bianco, i rischi di esplosioni, contaminazioni e, in altri casi (meno noti ma più diffusi di quanto non si voglia credere) radiazioni, malattie specifiche, inquinamenti, eccetera.

Come è noto, però, non è possibile affidare ad un C/64 (o ad altro elaboratore, pur se sofisticato) l'intera gestione di un complesso chimico, perchè è "cretino" (il C/64, ovviamente).

Nei casi estremi, comunque, è opportuno affidare ad "espertissimi" umani, notoriamente intelligenti e fantasiosi, il compito di invitare i dirigenti delle fabbriche (ma solo per cortesia e, soprattutto, se non disturba) a limitare la produzione oppure, nella impossibilità di farlo, almeno ad organizzare lotterie mettendo in palio bare in noce massiccio.

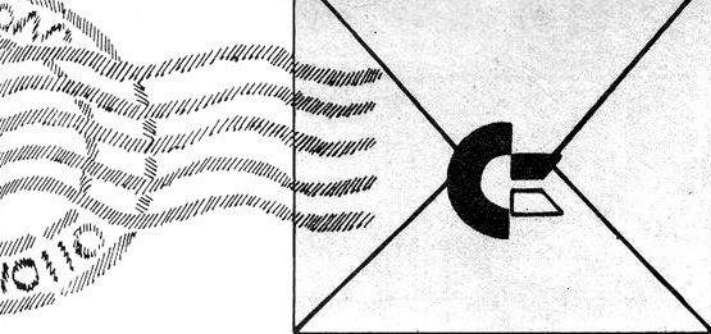
Dopo i puntuali disastri, inoltre, gli umani sono sempre in grado, al contrario di un semplice computer, di chiedere (sempre per cortesia) l'interruzione della produzione per alcuni mesi, aspettando che la gente si calmi e continui a sopportare.

A meno che i soliti facinorosi non si arrabbino sul serio, cosa che, finalmente, è avvenuta.

Spesso, poi, il problema viene risolto in modo così originale ed "umano" che il computer, di fatto, viene escluso nell'improbabile competizione intellettiva: innalzando, per decreto legge, i limiti minimi tollerabili.

Adesso rilassatevi in poltrona ed immaginate un fantastico mondo governato da cervelli al silicio in grado di interrompere non solo l'attività pericolosa di alcuni complessi industriali ma, addirittura, la stessa costruzione di fabbriche ad alto rischio.

Prima di abbandonarvi a simili fantasie, però, caricate la sveglia: i bei sogni non devono distrarvi dall'ascoltare i notiziari relativi alle intelligenti attività degli umani...

Alessandro de Simone

la vostra posta

AMIGA IN DISSOLVENZA

□ Posseggo un A-2000 e due videoregistratori; vorrei realizzare, su cassetta, dissolvenze incrociate.
(Massimo Monico - Casalpusterlengo)

• Il problema si può risolvere solo mediante apparecchiature specifiche per Vcr denominate, di solito, processori video e che devono consentire anche la miscelazione del segnale video.

Nel caso di immagini registrate, infatti, la miscelazione di due segnali non è semplice come nel caso di due segnali audio in cui sono sufficienti, al limite, un paio di transistor.

E' infatti necessario che il "quadro" dei due apparecchi video sia in perfetto sincronismo; questo problema viene risolto con un particolare circuito elettronico, denominato Genlock, caratteristica di tutti i (costosi) miscelatori.

Per l'Amiga esistono vari dispositivi Genlock (ed altri, piuttosto economici, dovrebbero esser presentati al SIM ed allo SMAU) ma consentono di miscelare il segnale del computer con quello di un videoregistratore o di una telecamera. Per effettuare dissolvenze incrociate, ed altri interessanti funzioni di questo tipo, sono quindi necessari specifici apparecchi.

PRIMI TENTATIVI

□ Ho scritto un programma per il mio C/128 che mi piacerebbe veder pubblicato.
(Luca Casadei - P.te Abbadese)

• Non ci è possibile pubblicare il listato che hai inviato per due motivi: il primo è perchè non lo hai spedito su disco (come raccomandiamo decine di volte su queste pagine). Il secondo motivo, invece, riguarda l'efficienza dello stesso programma, che contiene un numero eccessivo di istruzioni Circle.

Lo stesso effetto (riproduzione di numerosi cerchi concentrici) lo si può infatti ottenere con una sola istruzione che faccia ricorso al ciclo For...Next relativo all'incremento del raggio ed all'eventuale modifica della posizione del centro.

Tuttavia, a dispetto della complessità del programma inviato, devo riconoscere che di pazienza ne hai davvero tantissima e non ti fai scoraggiare dalla digitazione di lunghi listati. Ti consiglio, quindi, di digitare i programmi che pubblichiamo e di diventare padrone delle varie tecniche di programmazione che ti permetteranno, entro breve tempo, di diventare un vero esperto.

CP/M PER C/64

□ E' disponibile il sistema operativo CP/M per il Commodore 64?
(Luca Vicastri - Maserà PD)

• Alcuni anni addietro fu commercializzata una scheda, dotata di microprocessore Z/80, che consentiva di adattare il C/64 al popolare sistema operativo CP/M della Digital Research.

Purtroppo, però, non riuscii a procurarmela, perchè era piuttosto rara, e decisi di rinunciare del tutto alla sua ricerca dopo l'acquisto del C/128-D. Questo, come ben saprai, la contiene al suo interno.

Il CP/M può, infatti, funzionare solo su schede dotate di microprocessore Z/80 (o della stessa "famiglia") dal momento che adattarvi altri micro sarebbe troppo difficile. Se ti dovesse capitare di procurartene una, tuttavia, tieni presente che può funzionare solo con il drive: il registratore non può essere utilizzato in nessun caso.

Per ciò che riguarda la vendita dei nostri prodotti leggi le ultime pagine della nostra rivista.

BRAVO

□ Ho scoperto alcune regolette per la soluzione del problema "Le otto regine" (C.C.C. n.49).
(Aronne Bonomo - Vigonza)

• Non pensavo che quel programma (e relativo problema) ottenesse tanto successo; non è la prima volta, infatti, che un lettore ci "costringe" ad occuparcene.

Nonostante, sul numero scorso, siano state pubblicate numerose soluzioni, mi sento in obbligo di citare questo nostro bravo lettore che ritiene di aver scoperto alcune regolette fondamentali:

- Trovata una soluzione è spesso sufficiente "ruotarla" di 90 gradi (per tre volte) per ottenere altrettante soluzioni diverse.
- Spesso una soluzione "speculare" (da girare, eventualmente, ancora di 90 gradi) offre nuove soluzioni.
- Lasciando inalterati i numeri, ma "ruotando" le lettere, si ottengono altre soluzioni; ad esempio, la sequenza...

6A, 4B, 7C, 1D, 8E, 2F, 6G, 3H

...suggerisce la soluzione:

6H, 4G, 7F, 1E, 8D, 2C, 6B, 3A

Il lettore ha elencato altre soluzioni, scovate confrontando varie alternative molto simili o... molto diverse tra loro, ma che omettiamo di pubblicare suggerendo ai nostri lettori di scoprirle per proprio conto.

PUNIZIONI

□ Ad una mia precedente lettera, contenente due domande, avete dato una sola risposta; stavolta, per punizione, ve ne invio tre...
(Leda Litucci - Foggia)

• Diamine, la prossima volta starò più attento!

1- In un floppy disk, oltre ai byte relativi ai file programma (o sequenziali, USR, relativi, eccetera) esistono molti altri byte che svolgono la funzione di "puntatori" (una specie di promemoria). Ciò significa che il S.O. del drive deve sapere, in qualche modo, dove rintracciare i vari byte che, concatenati con altri, formano il programma o il file desiderato. Naturalmente l'utente non si accorge minimamente della gestione di questi puntatori: tutto ciò che "vede" è che, al momento del Load, il programma (il file) viene caricato (viene letto).

Se, però, si utilizza un particolare programma che è in grado di scandagliare il disco stesso, è possibile non solo individuare i singoli byte che compongono il programma, ma anche il contenuto dei puntatori di cui abbiamo parlato. Molte tecniche di protezione, anzi, si basano proprio sull'alterazione di questi puntatori, con vari effetti sui quali non ci intratteniamo.

Sul disco, però, oltre ai dati finora esaminati, esistono altre zone magnetiche contenenti informazioni, che possiamo definire hardware, necessarie per la gestione, appunto, h/w del disco stesso. Queste informazioni non possono essere esaminate con i vari Tool e sono piuttosto difficili da gestire. E' come se volessimo vedere i nostri occhi con i nostri occhi o prendere la nostra mano destra con la stessa mano destra.

Vi sono, tuttavia, numerose utility, sviluppate dai soliti geni, che sono in grado di accorgersi di varie anomalie relative a queste ulteriori informazioni contenute su ciascun floppy.

2- Non so dirti se la cartuccia citata funziona anche con il C/64 SX che possiedi. Ma perchè lo chiedi a me e non alla Niwa, che la produce e la vende?

3- Sulla traccia 18 settore zero sono memorizzati i dati relativi ai blocchi liberi (ed altre informazioni). Nel caso in cui un dischetto sia stato utilizzato più volte (cancellazione di file, registrazioni successive, cambiamento di nome, copia di file, operazioni di append, errori involontari di scrittura, eccetera) è probabile che rimanga traccia di tali informazioni. Per sapere esattamente che cosa succede alla directory di un disco, ti consiglio di prenderne uno nuovo, formattarlo con la ID (per esser sicuri che tutto venga accuratamente cancellato) ed effettuare varie prove, tra cui: registrare un programma due volte (con due nomi diversi) ed esaminare, con apposito Tool, che cosa è successo alla traccia 18; quindi cancellare il primo ed esaminare nuovamente la situazione. Poi registrare un file sequenziale senza dare il comando Close; e vedere che cosa sia successo; e così via di questo passo, accumulando esperienza dopo esperienza.

MISTERO RISOLTO

□ Ho provato, con il mio telefono, a formare il numero suggerito su C.C.C. n. 51 ma non sempre riesco a sentire la mia voce. Come mai?
(Simone Malaspina - Sestri)
(Ludovico Quercia - Cetraro)

• Un nostro lettore, di cui portroppo ho dimenticato il nome, aveva inviato tempo fa la risposta al quesito. Il numero di telefono, infatti, corrisponde ad un abbonato dell'Iran. Questa nazione, per i motivi bellici tristemente noti, spesso interrompe le comunicazioni internazionali per timore di attività spionistiche. Ne consegue che le centraline elettroniche a volte attivano, automaticamente, il consueto segnale di occupato, a volte "rinviano" alla fonte lo stesso segnale di chiamata (con il conseguente effetto "eco" di cui parlammo). Altre volte ancora la nostra stessa SIP, per evitare intasamenti di linee, attiva una voce preregistrata che, non appena viene formato il prefisso internazionale, avverte dell'impossibilità di effettuare il collegamento desiderato.

OBIEZIONE

□ Avrei un'obiezione (su Amiga) ed una domanda (per ciò che riguarda la formattazione del drive 1541).
(Dino Calvisi - L'Aquila)

• Il nostro atteggiamento nei confronti dell'Amiga (negativo sul N. 33, 34, ampiamente positivo dal 45 in poi) non è per nulla contraddittorio.

Nel 1986, infatti, eravamo abbastanza informati per sapere che il modello A-1000 avrebbe avuto un successo piuttosto limitato. I fatti, in seguito, ci hanno dato ampiamente ragione dal momento che il "vero" Amiga doveva essere il modello 500 che aveva maggior possibilità di diffusione (evento puntualmente verificato).

Nel 1986, data dei numeri 33 e 34

di C.C.C, volevamo mettere in guardia i nostri lettori da acquisti frettolosi. Chi ha seguito i nostri consigli ci sta ancora ringraziando.

Per ciò che riguarda la formattazione del 1541 oltre la 35ma traccia vi sono in giro numerosi programmi di utilità (molto complessi) che riescono nell'intento. Se, però, ti illudi di basare su tale sistema una tecnica di protezione, tieni presente che esistono altrettanti copiatori specifici: mettiamoci in testa, una buona volta, che non può esistere un sistema anti-copia!

PACKAGE DI DOMANDE

☐ Ho varie domande da fare...
(Fabio Bradach - La Spezia)

• ...ed ecco le risposte:

1- il drive è probabilmente disallineato. Prova a formattare un disco nuovo (cioè con la ID) e a registrare e leggere vari programmi. Se tutto funziona correttamente, il drive è da allineare perchè non "legge" gli altri dischi formattati in modo diverso (cioè... giusto); in caso contrario c'è qualche problema più serio.

2- E' possibile usare la User Port per collegare vari dispositivi in ingresso (deviatori, pulsanti, interruttori) ed in uscita (transistor, porte logiche). Un giorno, se avrò un po' di tempo, costruirò questa benedetta scheda e scriverò il relativo articolo per programmarla (che ho promesso da molti mesi).

3- Per ordinare un qualsiasi nostro arretrato, o il software di nostra produzione, è sufficiente mandarci il loro nome, il tuo indirizzo ed i tuoi soldi (scusa la brutalità...)

4- La routine l.m. del sistema operativo posizionata a partire da $AEFD è spesso utilizzata nelle routine proposte da G. Mariani nell'enciclopedia di routine; per comprenderne il funzionamento è sufficiente studiare con attenzione i disassemblati commentati che pubblichiamo in quella stessa rubrica.

DUBBI

☐ Vorrei comprare un drive per il mio C/64: il 1541 mi sembra lento (ma ricco di software), il 1581 è molto veloce (ma privo di programmi). Mi trovo in una situazione imbarazzante.
(Enzo - Torino)

• Confermo

MS/DOS E GW/BASIC

☐ Con il vostro Ms-Dos emulator non riesco a digitare i programmi che pubblicate.
(Paolo Semplicini - Firenze)

• Il disco di nostra produzione emula due "procedure". La prima (Ms/Dos Emulator) consente di usare il C/64 come se fosse un IBM compatibile e

VARIABILE ST

☐ A volte la variabile ST assume diversi valori. Quale è il loro significato?
(Aldo Provinzano - Palermo)

• La variabile ST è la variabile "riservata" degli home computer Commodore (Vic/20, C/64, C/16, Plus/4, C/128) che consente di verificare la presenza di eventuali anomalie nel corso di un colloquio con una periferica.

Nonostante ST abbia la "forma" di una variabile in vigola mobile, in effetti può contenere solo valori interi, per di più compresi tra 0 e 255. ST, infatti, contiene un'informazione per ciascun bit del byte che la rappresenta e cambia significato, in accordo con la seguente tabella, a seconda se il colloquio si realizza con il registratore o con il drive:

Registratore (fase di lettura o di verifica)

1: OK.
2: OK.
4: Il blocco dei dati letto contiene meno dati del previsto
8: Il blocco dei dati letto contiene più dati del previsto
16: Errore generico di lettura (o di verifica)
32: Errore di parità (alcuni bit non sono stati letti correttamente)
64: Fine del file
128: Fine del nastro

Drive

1: Ricezione non disponibile
2: Trasmissione non disponibile
4, 8, 16, 32: Errori non codificati
64: Fine del file
128: Drive non collegato

E' possibile che il valore associato a ST rappresenti una somma di più valori; in questo caso gli errori verificati sono tutti quelli corrispondenti ai codici che realizzano la somma stessa.

CAMBIAR CARATTERE

☐ E' possibile inserire variazioni al set di caratteri originali inserendo, ad esempio, la "e" accentata, la dieresi e così via?
(Mario Morniroli - Arma di Taggia)

• L'operazione è certamente possibile, grazie ad uno dei numerosi programmi di ridefinizione dei caratteri che abbiamo pubblicato più di una volta.

A parte la pazienza necessaria per modificare i caratteri desiderati, è necessario, però, tenere ben presenti alcune considerazioni.

La modifica del set di caratteri è "attiva" solo all'interno del computer; ne consegue che, ovviamente, è possibile visualizzarla su schermo. Tentando di inviare su stampante un messaggio, contenente caratteri modificati, si ha la sgradita sorpresa di vedere su carta i caratteri soliti, non modificati.

Ciò significa che bisogna programmare ANCHE la stampante per ottenere copie su carta. L'operazione, in questo caso, è lunga e complessa e porta ad elaborazioni di esasperante lentezza.

Se ciò non bastasse, tentando di caricare un programma professionale (come un w/p), dopo aver modificato il set di caratteri, questi vengono ripristinati come erano prima, annullando completamente la fatica impiegata.

Morale: per visualizzare su schermo caratteri insoliti è necessario usare programmi di propria creazione (o programmi che prevedano esplicitamente l'uso di set alternativi). Per stampare caratteri speciali è bene procurarsi una stampante che già li possegga (ed il software in grado di farli "venir fuori" senza eccessivi problemi!).

si limita, pertanto, ad offrire molte potenzialità del famoso sistema operativo.

L'altra procedura, invece, emula il linguaggio Gw/Basic con il quale è possibile digitare i programmi che vedi pubblicati.

Se ti limiti a caricare solo il primo file del disco, attivi l'emulatore Ms/Dos. Devi quindi caricare il linguaggio Gw-Basic seguendo le dettagliate e chiare istruzioni del manuale allegato alla confezione.

S/W PROFESSIONALE

☐ Secondo me dovreste affrontare le problematiche relative al corretto utilizzo dei potenti package come DB 3, Lotus 1-2-3 e simili.
(Carlo Morelli - Racconigi)

• I package citati dal nostro lettore sono, in effetti, programmi professionali di ottima fattura; purtroppo hanno il "torto" di non girare sul C/64, ma solo su computer IBM compatibili.

Per il momento sono pochi i lettori che posseggono tali elaboratori e non ci è possibile trattare argomenti ad essi relativi ad esclusione di articoli, più o meno generici, sui PC Commodore. In un prossimo futuro, chissà...

S/W FINANZIARIO

☐ Vorrei proporre un mio programma che consente di elaborare dati relativi alla borsa valori e al rendimento dei titoli azionari.
(Paolo Triberti - Alessandria)

• Non dubito della validità del software proposto, ma la maggior parte dei nostri lettori non gioca in borsa e non penso che il complesso programma possa interessare.

Semmai possono essere utili alcune delle routine che, ne sono certo, fanno parte del programma stesso: statistica, previsioni ed altre ancora, suscettibili di generalizzazione.

L'invio del software, tuttavia, deve essere effettuato esclusivamente su disco e non prima di aver concordato, per telefono, le modalità della collaborazione.

DIFETTI DI C.C.C.

☐ Quando pubblicate i vostri articoli, cercate di non dimenticare coloro che non posseggono "tutto" (Toma, Gw-Basic, eccetera) ma che dispongono del solo computer.
(Marco Cevoli - Brescia)

• In effetti il nostro simpatico lettore ha inviato quasi esclusivamente elogi nei nostri confronti; ho tuttavia preferito "estrarre" l'unico rimprovero presente nella sua lettera perchè è bene chiarire alcune cose.

L'informatica è una palestra in continua evoluzione. L'utente al quale ci rivolgiamo è (dovreste averlo capito) fondamentalmente curioso, prevalentemente cocciuto, costantemente intraprendente. Una persona così possiede certamente tutto il software di questo mondo (quello che manca alla sua collezione se lo procurerà domani). Non so quanti siano coloro che abbiano digitato le routine di Toma o abbiano acquistato il nostro emulatore Gw-Basic. Noi della Systems speriamo intimamente che, per ogni copia venduta, ne circolino almeno tre copiate; e questo non certo per soddisfare il nostro innato autolesionismo, ma perchè siamo convinti che l'informatica è cultura e questa DEVE diffondersi con ogni mezzo (pur-

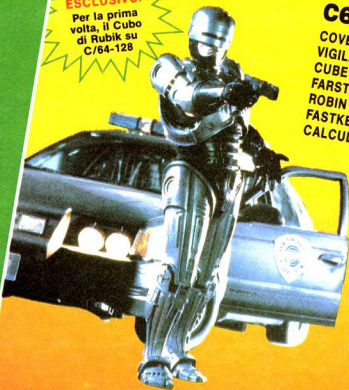
TUTTI VE

SEM

RSO IL DOS

MS-DOS

Personal computer

PRE UN MEGABYTE AVANTI

chè legale, tipo scambio tra amici SENZA fini di lucro).

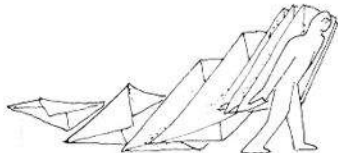
Se, pertanto, vi mancano alcuni programmi, procurateveli pure (almeno i nostri) in vari modi, senza andare tanto per il sottile. Oppure mettevi d'accordo in tre o quattro amici ed acquistate, presso il nostro servizio arretrati, una sola copia dei programmi che vi interessano, per poi duplicarveli con comodo, dal momento che non sono protetti (tranne che in alcuni, rarissimi casi).

La possibilità che noi offriamo, in modo così esplicito, pone quindi in secondo piano la difficoltà di usare package di vario tipo per la digitazione dei programmi che pubblichiamo.

COLORI IN GW-BASIC

□ Quando uso il vostro emulatore Gw-Basic, e tento di cambiar colore dopo aver attivato l'alta risoluzione, viene alterato anche il colore dei disegni già tracciati. Come mai?
(Alfonso Di Napoli - Baia Domizia)

• L'emulatore è principalmente uno strumento didattico e non professionale. Del resto il C/64 è un home computer e non è possibile pretendere di emulare le notevoli potenzialità grafiche di computer ben più potenti.

Ne consegue che, per ciò che riguarda il colore, bisogna accontentarsi delle potenzialità tipiche del C/64. Se avessimo introdotto la possibilità di modificare anche il colore, avremmo dovuto occupare molti altri Kbyte di memoria, lasciando, alla fine, ben poco all'utente finale. In ogni caso si tenga presente che con il C/64 non è possibile colorare con 16 colori diversi una matrice di 8x8 pixel, ma al massimo se ne possono sfruttare 4. E questo per motivi hardware legati alla struttura dello stesso C/64.

GEOS E CARTUCCE VUOTE

□ Vorrei sapere dove procurarmi cartucce in cui inserire Eprom da me programmate; inoltre ho un problema con il Geos.
(Claudio Masetti - Scandicci)

• Il prodotto richiesto è un po' particolare perchè, di solito, viene usato dalle industrie del settore (che se le procurano all'ingrosso) oppure da hobbysti, che le realizzano da sole.

Non appena verremo a sapere il nominativo di qualche ditta che vende tali cartucce vuote per corrispondenza, lo pubblicheremo immedia-

FUORI L'AUTORE

□ Digitando, con il C/128 ed il Plus/4, una particolare Sys, viene visualizzato il nome degli autori delle Rom. Esiste qualcosa di simile per il C/64?
(Michele Lo Console - Bari)

• No, che io sappia. Ne approfitto della segnalazione per ricordare che i programmatori non sono esseri aridi e privi di fantasia; lo dimostrano queste piccole "cose" che i progettisti, da un po' di tempo, si divertono ad inserire addirittura nelle Rom dei sistemi operativi!

RIBATTERE UN RIGO

□ Come posso fare, con la mia Mps-803, a ribattere due (o più) volte un rigo in modo da evidenziare alcune frasi?
(Franco Pardo - Ginosa)

• La stampante Mps-803 è una stampante economica e le potenzialità che offre sono in linea con il suo prezzo di vendita.

La possibilità di ribattere lo stesso rigo è presente in stampanti più "evolute" ed è attuabile via software, mediante l'invio di caratteri speciali, oppure, molto più spesso, mediante un pulsante (presente sul frontalino della stampante) che facilita l'inserimento manuale della comoda opzione. Questa prende, più propriamente, il nome di NLQ (Near Letter Quality = qualità simile a quella di una macchina da scrivere) e si esplica "ripassando" due volte ciascun rigo, sfalsando la testina di stampa (al secondo passaggio) di alcuni centesimi di millimetro in modo da rendere meno evidente la matrice di punti generata dagli aghi della stampante.

Il sistema NLQ si limita a ribattere due volte lo stesso rigo. Se la frase, nonostante tale accorgimento, non risulta evidente, è necessario cambiare il nastro inchiostrato. Pur se in taluni casi, via software, è possibile ripassare più di due volte la frase desiderata, l'operazione è da evitare perchè si rischia di perforare il foglio di carta.

9 E 24 AGHI

PERIFERICHE

□ Usando il Geos 128 con una stampante a 24 aghi mi accorgo che, in realtà, questa stampa con la qualità dei 9 aghi. Come posso rimediare?
(Gerardo Canoro - Lucca)

• Un qualsiasi package per C/64 o C/128 prevede, di solito, l'uscita su stampante Mps-803. Se la periferica collegata risulta di altro tipo (ma compatibile 803), questa viene "configurata", cioè resa simile ad una 803.

Questo può essere uno dei motivi della limitazione riscontrata.

Un altro motivo è legato al fatto che, in fase di stampa, viene "scaricata" la pagina grafica e non la pagina testo. In altre parole, ciò che viene visualizzato su schermo sembra essere una normale pagina testo ma, in realtà, è la pagina grafica in cui sono stati riprodotti i vari caratteri alfanumerici.

In questo caso (hard-copy di hi-res) molte stampanti non funzionano più in modo ottimale (24 aghi) e si limitano a riprodurre la schermata come una qualsiasi altra stampante di più modeste caratteristiche.

In entrambi i casi, purtroppo, non c'è nulla da fare.

tamente, per la gioia dei nostri lettori smanettoni.

Per ciò che riguarda la versione di Geos in tuo possesso (che non "riconosce" il mouse originale Commodore) ritengo che si tratti di una copia mal effettuata e, comunque, non originale. Ti consiglio, ovviamente, di procurartene una efficiente.

SOVRAPPORRE DUE PAGINE

□ Come posso visualizzare due pagine hi-res in modo che, facendo scrollare la prima, appaia la seconda e viceversa?
(Paolo Bevilacqua - Sciacca)

• E' necessario che le le due pagine grafiche siano presenti contemporaneamente in memoria; inoltre è necessario che sia presente anche la pagina colore, relativa alla seconda schermata, da richiamare con tecniche opportune al momento della visualizzazione. Se ciò non bastasse è necessario ricorrere massicciamente al Raster Register.

L'ipotetico programma in grado di svolgere la funzione richiesta dovrebbe funzionare in questo modo:

- Settare i puntatori della prima pagina e visualizzarla per il numero di pixel (verticali) desiderati, che supponiamo sia 150.
- A partire dal 151mo pixel deve essere visualizzata la seconda pagina grafica (dopo aver alterato i puntatori hi-res) che deve avere anche gli attributi colore ad essa relativa; lasciando "attiva" la stessa pagina colore si ottiene un disegno confuso e comunque diverso da quello originario (a meno che entrambi i disegni siano tracciati in due soli colori per l'intero schermo).
- arrivati al 200mo pixel (l'ultimo in basso), il ciclo riprende dall'inizio dopo aver nuovamente modificato i puntatori di pagina grafica attiva.

La presenza contemporanea di due pagine grafiche, che occupano 16 KRam, costringe il programmatore, di solito, a fare salti mortali per la loro corretta gestione, e questo nell'eventualità in cui si desideri visualizzarle, una alla volta, per intero. Uno sforzo notevolmente maggiore richiede, poi, lo scroll di una sull'altra e viceversa, per non parlare della velocità di elaborazione e degli inevitabili tremolii che possono verificarsi durante l'alterazione dei puntatori.

COLLEGAMENTI

PERIFERICHE

□ E' possibile collegare due computer C/128 tra di loro senza ricorrere alla User Port?
(Bartolo Scarafia - Genova)

• La User Port è l'unica porta che consente di effettuare facili scambi di dati non solo tra due C/128 ma tra il C/128 (o C/64) e qualsiasi altro computer dotato di interfaccia Rs-232.

Si potrebbe obiettare che esiste un'altra porta seriale a disposizione, e cioè quella utilizzata per il collegamento con il drive e la stampante. Tale connettore, purtroppo, presenta diversi problemi di varia natura che ne sconsigliano l'utilizzo.

C'è però un altro sistema per scambiare dati tra due computer con una certa facilità (e, soprattutto, rapidità).

E' sufficiente collegare (quando TUTTE le apparecchiature sono SPENTE) i due computer ad un unico drive. In seguito, dopo aver acceso i tre apparecchi, uno dei due computer "scaricherà" dati o programmi sul disco; terminata tale operazione toccherà al secondo computer la lettura di quanto registrato. In pratica si tratta di usare il drive come buffer temporaneo di dati. L'importante è che, mentre un computer utilizza il drive, l'altro non vi acceda, pena l'inchiodamento del sistema.

risposte rapide

REGISTRARE PRIMA DI SPEGNERE
(Matteo Marchini - Marmirolo)

Dopo aver digitato un programma è indispensabile registrarlo su supporto magnetico (nastro o disco) perchè spegnendo il computer qualunque dato viene cancellato. Ti consiglio di leggere con attenzione il libretto di istruzioni del tuo calcolatore perchè l'operazione di salvataggio (Save) e di caricamento (Load) sono le più importanti.

VIC EMULATOR
(G. Morino - Sestri Levante)

Non sono a conoscenza di un emulatore di Vic/20 per C/64 che consentirebbe, con quest'ultimo elaboratore, di far girare i programmi idonei per il vecchio computer Commodore. Tuttavia ritengo che, al giorno d'oggi, non ha senso utilizzare un simile prodotto perchè in tutti i campi del software (dai video game ai professionali) esistono prodotti specifici per C/64 di gran lunga migliori degli analoghi prodotti per Vic/20.

COMPILATORE GRAFICO
(Filippo Oliva - Misterbianco)

Il "compilatore grafico matematico", di cui spesso parliamo, è un nostro prodotto su cassetta (da richiedere al servizio arretrati) che permette di tracciare, ad alta velocità, funzioni matematiche complesse.

POSSIAMO VEDERLO
(Navid Anayati - Brugherio)

Il software realizzato sembra interessante. Telefonaci e fissa un appuntamento per esaminarlo da vicino.

DOVE E' L'ERRORE?
(Giuseppe Bove - Napoli)

Purtroppo non dici in quale linea il computer segnala l'errore e non sono in grado di aiutarti.

1541 VERTICALE
(Simone Malaspina)

Non vi sono controindicazioni sul posizionamento del drive che, quindi, puoi tenere appoggiato sul fianco. E' tuttavia necessario accertarsi che vi sia un'ottima circolazione dell'aria: la posizione sul fianco, infatti, potrebbe impedirne il regolare ricambio a causa del diverso orientamemto delle fessure di areazione.

IMPOSSIBILE
(Aldo Provinzano - Palermo)

Non è possibile memorizzare, con un C/64, l'immagine proveniente da un apparecchio TV dotato di uscita video, a meno che non utilizzi l'apposito accessorio (digitalizzatore TV) descritto sul N.54 e venduto dalla ditta Niwa di Sesto San Giovanni (Tel. 02/26.20.312).

MERLIN PER C/128
(Maurizio Romboli - Bologna)

Il programmatore di Eprom "Merlin", decritto sul n.53, funziona solo sul C/64 oppure sul C/128 in modo 64. Ma che importanza ha, mi chiedo, se non funziona in modo 128? L'importante è che si riesca a programmare le Eprom...

SOLO SU DISCO
(Simone Pistolesi - Siena)

Purtroppo non posso esserti di aiuto perchè non hai inviato il listato su disco e non mi è assolutamente possibile rintracciare un errore nella marea di dati contenuti nel listato inviato su carta.

80 COLONNE
(Giuseppe Di Molfetta - Bari)

Lo schema per far apparire su un comune TV (o monitor) le 80 colonne del C/128 è stato già pubblicato sul N.29 di Commodore Computer Club.

RITARDI
(Marco Pizzo - Palermo)

Il nostro lettore protesta per il notevole ritardo con cui il centro specializzato Commodoctor della sua città ha riparato il suo Amiga.

SOCCER AL LASER
(da alcune telefonate)

Il dischetto Supersoccer (per computer IBM compatibili) distribuito in edicola, contiene una speciale protezione al laser che ne impedisce la copia. L'unico modo per rimuovere la protezione è quello di rintracciare, a livello di linguaggio macchina, le routine preposte al controllo, e di rimuoverle. A patto di esserne capaci.

CARTA RICICLATA
(da alcune telefonate)

La carta con cui si realizza l'inserto di C.C.C. (come TUTTI gli inserti della Systems editoriale) è riciclata ed è questo il motivo per cui la sua qualità appare più modesta rispetto a quella delle altre pagine. Riteniamo, tuttavia, che i nostri lettori siano sensibili al problema ecologico e apprezzino la decisione di utilizzare, almeno in parte, un tipo di carta più rispettosa dei problemi ambientali.

AMIGAZZETTA
(da alcune telefonate)

Il dischetto "Amigazzetta", su cui torneremo ampiamente nei prossimi mesi, ha una periodicità mensile ed è venduto solo per corrispondenza al prezzo di L. 15000, comprese le spese di spedizione. Per maggiori informazioni telefonate al servizio arretrati (Tel. 02/84.67.348).

MESSAGGERO VENETO

Con vero piacere notiamo che il quotidiano "Messaggero Veneto" dedica ampio risalto ai videogame più recenti ed ai periodici di informatica, tra cui i nostri, di cui fornisce, con precisione e competenza, una breve descrizione dei contenuti.

UN NUOVO VESTITO PER I TASTI DEL C/128

Come ridefinire i caratteri del C/128 utilizzando la pagina grafica del potente computer

di **Michele Galante**

La possibilità di ridefinire il set di caratteri si può rivelare utile in molte situazioni, come per esempio nei videogiochi oppure per creare caratteri personalizzati.

Il C/128 possiede un chip video (VIC 2) quasi uguale a quello del C/64, con il quale è, comunque, perfettamente compatibile. Grazie ad un registro di questo chip possiamo ridefinire il set di caratteri del C/128.

LA TEORIA

Come molti sapranno, esiste una ROM nel nostro computer, chiamata ROM generatrice di caratteri, nella quale sono memorizzati stabilmente tutti i dati ad essi relativi. Più precisamente, i set di caratteri sono due, selezionabili mediante la pressione contemporanea dei tasti Shift e Commodore. La ROM di cui parliamo si trova nel banco 14 dall'indirizzo $D000 a $DFFF. La prima metà ($D000-$D7FF), contiene il set Maiuscolo / Grafico che è quello che troviamo all'accensione del computer; la seconda metà ($D800-$DFFF), invece, contiene il set Maiuscolo / Minuscolo, che è possibile selezionare premendo i tasti sopra citati.

Potete facilmente notare che ogni set occupa 2048 byte perchè i caratteri memorizzati sono 256 ed i byte necessari per definire ciascun carattere sono 8 (8*256 = 2048).

Cerchiamo ora di capire come il

SCHEDA TECNICA

Software per applicazioni grafiche

Hardware richiesto: C/128; non adattabile ad altri computer Commodore

Anche il programma pubblicato in queste pagine (oltre al alcuni file applicativi) è contenuto nel disco "Directory" di questo mese.

Bit	7654	3210		Num. dec.
Byte 1	0011	1100	=	60
Byte 2	0110	0110	=	102
Byte 3	0110	1110	=	110
Byte 4	0110	1110	=	110
Byte 5	0110	0000	=	96
Byte 6	0110	0010	=	98
Byte 7	0011	1100	=	60
Byte 8	0000	0000	=	0

Fig. 1

computer interpreta questa marea di dati. E' risaputo che un byte contiene 8 bit che possiamo interpretare come punti. Un bit posto a zero è un punto spento, un bit posto ad uno (settato) è invece un punto acceso sul video. Con 8 byte possiamo quindi rappresentare una griglia di 8*8 punti per un totale di 64 punti (= bit). Accendendo e spegnendo i punti di tale griglia potremmo disegnare qualsiasi carattere.

Riferendoci alla figura 1, trasformiamo i numeri binari della griglia in numeri decimali e poniamoli nelle locazioni di memoria interessate. I byte vengono memorizzati sequenzialmente, cioè uno dietro l'altro, in modo che le locazioni $D000-$D007 contengano gli 8 byte del primo carattere, $D008-$D00F quelli relativi al secondo e così via.

Adesso sappiamo come il computer interpreta i dati della ROM di caratteri, ma come facciamo a ridefinirli? Una ROM, infatti, permette solo di essere letta (ROM = Read Only Memory, memoria a sola lettura) e non possiamo scriverci nulla; ne consegue che i caratteri ridefiniti dovranno per forza essere memorizzati in RAM.

A questo punto entra in gioco il registro del VIC 2 nominato all'inizio dell'articolo. Precisamente si tratta della locazione 53272 i cui 4 bit bassi (numerati da 0 a 3) indicano dove si trovano i dati del set di caratteri. Ma se provate a modificare il contenuto di questa locazione succede qualcosa di strano, anzi: non succede proprio nulla. Se controllate mediante Peek, vi accorgerete che non è possibile modificare la locazione 53272; ma allora questo registro del VIC 2 è su ROM?

No, la causa di questo fenomeno è un'altra. Sul C/128 (a differenza del C/64) i registri di alcuni chip vengono controllati dall'interrupt. In pratica se scriviamo qualcosa in queste locazioni, 1/60mo di secondo più tardi l'interrupt ripristina il precedente valore. Potrete capire meglio quanto detto se seguite l'esempio:

Entrate in modo 64 e digitate i comandi...

```
POKE 53269,1: POKE 53287,1
POKE 53248,150: POKE 53249,150
```

I lettori più svegli avranno capito che in questa maniera si accende lo sprite 1, lo si colora di bianco e lo si posiziona in un punto visibile dello schermo.In modo 128, digitando gli stessi comandi, dovrebbero accadere le stesse cose di prima; invece ci accorgiamo che non è possibile muovere lo sprite con gli stessi registri del VIC usati in modo 64, mentre usando MOVSPR riusciamo a spostarlo a piacimento.

Val.	Locazione
16	$0000-$0FFF
20	$1000-$1FFF (default)
24	$2000-$2FFF
28	$3000-$3FFF

Fig. 2

Se premiamo i tasti Run/Stop e Restore, e digitiamo nuovamente i comandi prima visti, vedremo che lo sprite appare per un istante per poi tornare subito alla posizione iniziale (cioè scomparendo alla vista) dopo un 60mo di secondo. Dobbiamo quindi concludere che è impossibile usare i registri sul C/128?

Prima di perdere ogni speranza, provate a digitare:

```
POKE 4566,150: POKE 4567,150
```

Adesso lo sprite si comporta in modo corretto. Resta da scoprire il perchè di un simile comportamento.

I REGISTRI OMBRA

Le due locazioni 4566 e 4567 prendono il nome di "registri ombra" e sono le locazioni da cui il ciclo di interrupt preleva i dati da immettere nei registri "veri". Il registro ombra relativo alla locazione 53272 è 2604.

Nella locazione 2604 va quindi introdotto uno dei valori riportati in figura 2, a seconda della zona in cui sono memorizzati i caratteri.

Si consiglia di memorizzare il set di caratteri ridefinito (in tutto o in parte) a partire da $2000 oppure $3000 dopo aver impartito i comandi Graphic 1: Graphic 0 per ottenere due scopi: il primo è quello di allocare gli stessi caratteri nella pagina grafica; il secondo consente di spostare l'inizio del Basic a partire da $4000. Esiste, a onor del vero, anche un modo diverso per memorizzare il set in altre locazioni, vale a dire cambiando il banco video del VIC. Questa tecnica, tuttavia, comporta alcuni problemi non facilmente risolvibili.

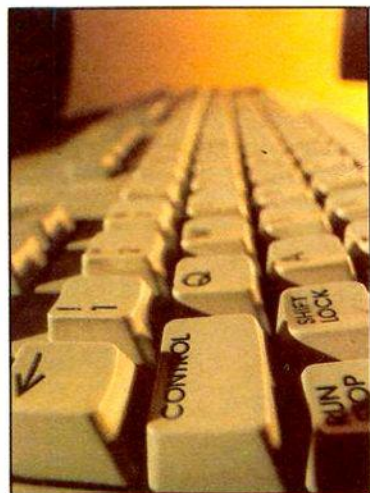
I lettori più esperti, inoltre, potranno sfruttare l'insolita allocazione di memoria (ci riferiamo alla pagina hires) per tracciare disegni nella pagina grafica che, alla fine, potrà essere considerata "anche" come set di caratteri ridefiniti.

```
01300  LDX #$00     Azzera il
01302  LDY #$00     contatore.
01304  LDA ($FA)    legge un byte
01306  STA ($FC)    e lo copia.
01308  CPY $FE      controllo byte basso
0130A  BEQ $1317    se uguale vai a 1317
0130C  INY          altrimenti incrementa
0130D  BNE $1304    contatore.
0130F  INC $FB      incrementa bytes alti
01311  INC $FD      delle locazioni
01313  INX          e del contatore.
01314  JMP $1304    continua loop.
01317  CPX $FF      controllo byte alto
01319  BNE $130C    se diverso continua loop.
0131B  RTS          altrimenti esci.
```

Fig. 3

IL PROGRAMMA

Il programma di queste pagine permette di creare e gestire con grande facilità propri set di caratteri che potranno poi essere usati in altri programmi. E' consigliabile compilarlo (con Pet Speed 128 o altri compilatori) a causa della lentezza del Basic.

Sul dischetto "Directory" di questo mese sono inseriti, infatti, sia il programma sorgente che quello compilato, oltre a due set di caratteri già definiti.

Dopo aver dato il RUN, il programma chiede se si desidera caricare un font già memorizzato su disco sul quale, ovviamente, si desidera apportare ulteriori variazioni.

Quindi si introduce il nome del nuovo font che si intende creare. Questo apparirà sullo schermo, in alto, oltre al set (1 oppure 2), al carattere con cui stiamo lavorando ed alla locazione che il carattere occupa in memoria. Nel centro appare la griglia con la "figura" da elaborare e, accanto, i valori dei singoli byte sia in decimale che in esadecimale.

La figura può essere modificata grazie al cursore (a forma di croce) che si comanda con i tasti cursore. Per accendere e spegnere i singoli punti si devono usare i tasti più (+) e meno (-); per accendere e spengere una riga basterà premere i tasti 1 e 2.

Home sposta il cursore nell'angolo superiore sinistro della griglia; Shift e Clr/Home cancella l'intera figura. Con F1 si può copiare uno (o più) caratteri in altra zona, con F3 possiamo visualizzare un altro carattere, con F5 viene effettuata la stampa su carta (oltre ai dati relativi), mentre con F7 si termina il lavoro e si può registrare l'intero set.

Con il tasto ESC vengono visualizzati tutti i caratteri del set, cosa che si ottiene molto facilmente in quanto i caratteri sono memorizzati, come già detto, nella pagina grafica.

Per utilizzare i caratteri elaborati con il programma proposto occorre caricarli in memoria dopo aver spostato l'inizio del BASIC alla fine della pagina grafica dal momento che i caratteri vengono memorizzati da \$2000 a \$2FFF. Quindi bisogna pokare il valore 24 in 2604 ed il gioco è fatto:

Graphic 1: Graphic 0:
Bload"Nome font.fnt":
Poke 2604,24

Ricordate che i set vengono memorizzati automaticamente con il suffisso ".FNT". Può rivelarsi utile avere subito a disposizione il font relativo al set di caratteri standard del C/128. Per realizzarlo seguite questo procedimento: entrate nel MONITOR del linguaggio macchina con SHIFT e F7; ricopiate la ROM generatrice di caratteri in RAM a partire da \$2000 con T ED000 EDFFF 2000; salvate il font su disco con S "STANDART.FNT", 8, 2000, 3000. Se inoltre possedete la versione del C/128 con tastiera italiana attivatela premendo il tasto CAPS LOCK e rieseguite le stesse operazioni descritte prima salvando il nuovo font con un nome tipo ITALY.FNT.

VIVA IL REGISTRATORE

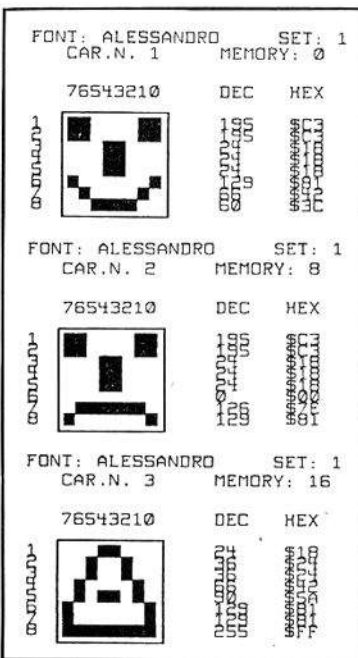
Un suggerimento per chi possiede il solo registratore a cassette (oltre al banale: comprate un drive!).

Sostituite il Graphic 1,1 di riga 180 con un semplice Graphic 1 e usate il MONITOR del linguaggio macchina per salvare...

S"nome", 1, 2000, 2FFF

...e per caricare i set...

L"nome"

Il programma fa uso di una piccola routine in linguaggio macchina che serve esclusivamente per eseguire il comando COPIA del programma.

```
100 REM RIDEFINIZIONE DEI CARATTERI PER C/128
110 REM BY MICHELE GALANTE
120 :
130 COLOR4,12:PRINT"▯":FAST
140 FORA=4864TO4891:READX:POKEA,X:NEXT
150 FORA=1TO4:KEYA*2-1,CHR$(A):NEXT
160 SLOW:CHAR,9,10,"*** FONT MANAGER ***",1:CHAR,18,12,"BY"
170 CHAR,9,14,"MICHELE GALANTE 1988":GETKEYA$
180 FAST:GRAPHIC1,1:GRAPHIC0:FORA=3584TO3648:POKEA,0:NEXT
190 PRINT"▯▮▮COSA VUOI FARE ?":PRINT"▮▮1 CARICARE UN FONT"
200 PRINT"▮2 CREARE UN FONT":SLOW
210 GETKEYA$:ONVAL(A$)GOTO230,250
220 GOTO210
230 INPUT"▯▮NOME DEL FONT (12 CAR.)";F$:IFLEN(F$)>12THEN230
240 BLOAD(F$+".FNT")
250 INPUT"▯▮NOME NUOVO FONT (12 CAR.)";F$:IFLEN(F$)>12THEN250
260 S=0:NC=0
270 PRINT"▯FONT: "F$:PRINT"SET:"S+1;TAB(13)"CAR.N.";
280 PRINT NC+1;TAB(26)"MEMORY:"(NC+S*256)*8
290 PRINTTAB(3)"▮▮76543210      DEC   HEX":PRINT"  ┌──────────┐"
300 FORA=1TO8:PRINTA;"▮||"SPC(8)"|":NEXT:PRINT"  └──────────┘"
310 CHAR,30,6,"╭─▮▮▮▮|| |▮▮▮▮|╰─╯":SPRITE1,1,2:MOVSPR1,272,106
320 CHAR,0,20,"▮F1▮  COPIA":CHAR,20,20,"▮F3▮  CAMBIA CARATTERE"
330 CHAR,0,22,"▮F5▮  STAMPA":CHAR,20,22,"▮F7▮  FINE"
340 CHAR,7,24,"▮ESC▮  VISUALIZZA CARATTERI"
350 FORA=0TO7:X=PEEK(8192+(NC+S*256)*8+A):FORB=1TO8
360 IF(XAND2↑(8-B))=0THENPOKE1266+A*40+B,32:ELSEPOKE1266+A*40+B,160
370 NEXT:NEXT
380 FORNR=0TO7:GOSUB910:NEXT
```

Dopo Hacker Cartridge & O.M.A.
NIWA è lieta di annunciarVi che finalmente è disponibile l'attesissima

"NIKI" CARTRIDGE

O.M.ALFRED & NIKI in collaborazione "esplosiva" hanno creato questo Hardware incredibile che è la sintesi dell'esperienza acquisita in tutti questi anni di lavoro sul Commodore 64 ed è la logica conseguenza di Hacker e O.M.A.

PREZZO 99.000 IVA INCLUSA

"NIKI" è la cartuccia rivoluzionaria che ti permette di fare oggi quello che le altre non faranno mai!!!

"NIKI" non è solo un imbattibile sprotettore ma molto di più:

- **Copia in un solo file,** indipendente dalla cartuccia, ogni programma che gira in memoria essendo completamente invisibile ad ogni tipo di software.
- **Super veloce**: in meno di un minuto copia un programma di 220 blocchi!!! Doppia velocità con il nastro e con il disco, da tre a 10 volte più veloce di tutte le altre Cartridges.
- **Super compatto**: tecniche intelligenti e avanzatissime compattano il programma in un solo file (salva più di tre programmi per facciata di disco).
- **Facile da usare**: tutte le funzioni si scelgono da menù non necessita disco con software.
- **Controllo degli sprite**: uno sprite monitor ti permette di vedere, salvare, cambiare gli Sprite e personalizzare così i tuoi giochi.
- **Hard Copy del video**: salva ogni videata Multicolor, compatibile Koala, Blazing Paddles, Graphic Slide Show.
- **Fast Loader per Disco**: carica 5 volte più veloce del normale e non occupa memoria (per una perfetta compatibilità).
- **Monitor incorporato**: per guardare ogni programma in memoria, i registri ed ogni cosa che ti serve.
- **Potente Toolkit**: include comandi come Old, Merge, Linesave, Append, Copy, ecc...
- **Tasti Funzione**: predefiniti per veloci operazioni sui comandi più usati (come list/run/directory...)
- **Nuovi comandi monitor**: monitor esteso con possibilità di dare comandi usando la sintassi del Basic (Blank/switching/ecc...)
- **File copy fino a 247 Blocchi**: file copy fino a 44 programmi, file user e sequenziali, in modo multicopie e supporta 2 Disk Drive (8 & 9)
- **Fast Save & Fast Format**: salva in modo turbo e formatta in 10 secondi.
- **Compatibile Speeddos**: permette di sfruttare al 100% i vantaggi del trasferimento dei dati in parallelo dovuti allo Speeddos.
- **Compatibile con Commodore** 64/64C, 128/ 128D (in modo 64), 1541/1541C/1570/1571, Speeddos/Turbo ROM varie.
- **Invisibile al sistema**: speciali tecniche rendono tutte le funzioni INVISIBILI al computer e quindi la riuscita del risultato è pressoché totale!

NESSUNO TI PUO' DARE DI PIU',

ed è per questo che questa Cartuccia porta il nome di **NIKI** la ragazza che ha rivoluzionato il mercato dell'Home Computer in Italia creando la NIWA. **NIKI** è più potente, ha più utilities, copia più programmi scavalcando qualsiasi schema di protezione.
È facilissima da usare basta inserirla nel Computer e premere un tasto.
È assolutamente invisibile e ti permette di avere il completo controllo sul Computer.
Da nastro a nastro, da disco a disco, da disco a nastro e da nastro a disco.
Tutti sono in grado di usarla perché non è richiesta nessuna esperienza, **NIKI** ti dice esattamente cosa devi fare in modo chiaro.

Diventa invulnerabile nei giochi con lo Sprite Killer!!! Visualizza, salva e carica gli Sprite da un gioco all'altro.

NIKI È TUTTO QUESTO E MOLTO DI PIU'. BISOGNA PROVARLA PER CREDERCI!

NIWA Hard & Soft — 20099 Sesto S. Giovanni (MI) — Via Bruno Buozzi, 94 — Telefono (02) 26.20.312

```
390 X1=0:NR=0:GOTO410
400 POKEL1,PEEK(L1)-11
410 LO=1267+NR*40+X1:POKELO,PEEK(LO)+11:L1=LO
420 GETKEYA$:IFA$="▯"ANDNR>0THENNR=NR-1:GOTO400
430 IFA$="▮"ANDNR<7THENNR=NR+1:GOTO400
440 IFA$="▮"ANDX1>0THENX1=X1-1:GOTO400
450 IFA$="▮"ANDX1<7THENX1=X1+1:GOTO400
460 IFA$="+"THENPOKELO,160:GOSUB910:POKELO,171
470 IFA$="-"THENPOKELO,32:GOSUB910:POKELO,43
480 IFA$="1"THENCHAR,3,NR+6,"         ":GOSUB910:POKELO,43
490 IFA$="2"THENCHAR,3,NR+6,"         ",1:GOSUB910:POKELO,171
500 IFA$="▮"THENX1=0:NR=0:GOTO400
510 IFA$="▯"THENWINDOW3,6,10,13,1:WINDOW0,0,39,24:GOTO380
520 ONASC(A$)GOTO550,750,780,880
530 IFASC(A$)=27THENGRAPHIC1:SPRITE1,0:GETKEYA$:GRAPHIC0:SPRITE1,1
540 GOTO420
550 WINDOW0,20,39,24:PRINT"▯INIZIO ZONA DA COPIARE:"
560 INPUT"▮SET (1-2) ";S2:IFS2<1ORS2>2THENPRINT"▯▯▯":GOTO560
570 INPUT"CARATTERE N. (1-256) ";C2:IFC2<1ORC2>256THENPRINT"▯▯":GOTO570
580 S1=S2-1:C1=C2-1
590 PRINT"▯FINE ZONA DA COPIARE:"
600 INPUT"▮SET (1-2) ";S2:IFS2<S1+1ORS2>2THENPRINT"▯▯▯":GOTO600
610 INPUT"CARATTERE N. (1-256) ";C2
620 IFC2<1ORC2>256OR(S2=S1+1ANDC2<C1+1)THENPRINT"▯▯":GOTO610
630 S2=S2-1:C2=C2-1
640 PRINT"▯INIZIO ZONA DESTINAZIONE:"
650 INPUT"▮SET (1-2) ";S3:IFS3<1ORS3>2THENPRINT"▯▯▯":GOTO650
660 INPUT"CARATTERE N. (1-256) ";C3:IFC3<1ORC3>256THENPRINT"▯▯":GOTO660
670 S3=S3-1:C3=C3-1
680 L1=8192+(C1+S1*256)*8:L2=8192+(C2+S2*256)*8:L3=8192+(C3+S3*256)*8
690 L2=L2-L1+7
700 IFL3+L2>12288THENL2=12288-L3
710 POKE250,L1-INT(L1/256)*256:POKE251,INT(L1/256):POKE252,L3-INT(L3/256)*256
720 POKE253,INT(L3/256):POKE254,L2-INT(L2/256)*256:POKE255,INT(L2/256)
730 SYS4864:WINDOW0,0,39,24
740 GOTO270
750 WINDOW0,20,39,24:INPUT"▯SET (1-2) ";S:IFS<1ORS>2THEN750
760 INPUT"▮CARATTERE N. (1-256) ";NC:IFNC<1ORNC>256THENPRINT"▯▯":GOTO760
770 S=S-1:NC=NC-1:WINDOW0,0,39,24:GOTO270
780 XX=NR:OPEN1,4:PRINT#1," FONT: ";F$;"     SET:";S+1:PRINT#1,"    CAR.N.";
790 PRINT#1,NC+1;"    MEMORY:";(NC+S*256)*8:PRINT#1
800 POKELO,PEEK(LO)-11
810 PRINT#1,"     76543210     DEC   HEX":PRINT#1,"     ┌────────┐";CHR$(8)
820 FORA=1TO8:PRINT#1,CHR$(15);A;"|";:FORB=0TO7
830 IFPEEK(1267+(A-1)*40+B)>128THENPRINT#1,"▮ ■";:ELSEPRINT#1," ";
840 NEXT:PRINT#1,"|";:NR=A-1:GOSUB910:X$="    ":MID$(X$,1)=STR$(X)
850 PRINT#1,"   ";X$;"   $";RIGHT$(HEX$(X),2);CHR$(8):NEXT
860 PRINT#1,CHR$(15);"    └────────┘":PRINT#1:CLOSE1:NR=XX
870 POKELO,PEEK(LO)+11:GOTO420
880 PRINT"▯SEI SICURO (S/N) ?":GETKEYA$:IFA$<>"S"THEN270
881 SPRITE1,0:PRINT"▯REGISTRI IL FONT (S/N) ?":GETKEYA$:IFA$="N"THEN900
890 BSAVE(F$+".FNT"),P8192TOP12288
900 RUN180
910 X=0:FORB=0TO7:V=PEEK(1267+NR*40+B):IFV=160THENX=X+2↑(7-B)
920 NEXT:CHAR,16,6+NR,"   "
930 CHAR,15,6+NR,STR$(X):CHAR,22,6+NR,"$"+RIGHT$(HEX$(X),2):POKE3584+NR*3,X
940 POKE8192+NR+(NC+S*256)*8,X:RETURN
950 DATA162,000,160,000,177,250,145,252,196,254,240,011,200,208,245,230,251
960 DATA 230,253,232,076,004,019,228,255,208,241,096
970 END
```

UN INSOLITO MANGIA-CASSETTE MUSICALE

Forse il programma che presentiamo non accontenterà gli appassionati di musica stero Hi-Fi; tuttavia, tenendo presente che bastano poche righe Basic...

di **Lorenzo Emilitri**

Se si dovesse svolgere un'inchiesta sulla periferica di I/O (Input Output) più venduta per il C/64, al primo posto figurerebbe il registratore a cassette; non certo perchè rappresenta il meglio come periferica, ma solo perchè la maggior parte delle confezioni del C/64 erano corredate con tale primordiale accessorio.

Purtroppo il sistema di registrazione su nastro non è esattamente il top, per ciò che riguarda la velocità, ed è questa la causa che ha generato l'enorme sviluppo dei Turbo Tape ed il successo dei drive, originali o compatibili.

Il Datassette non è solo in grado solo di salvare e, successivamente, caricare a velocità più o meno elevate programmi e dati, ma anche di ascoltare un segnale qualsiasi presente su nastro, proprio come un comune registratore. Tempo fa, su queste stesse pagine, fu presentato un argomento simile, ma stavolta presentiamo un programma più sofisticato.

SCHEDA TECNICA

Software per applicazioni grafiche

Hardware richiesto: C/64; impossibile da adattare ad altri computer Commodore

Richiede registratore

Consigliato ai principianti

Anche il programma pubblicato in queste pagine è contenuto nel disco "Directory" di questo mese.

UN PO' DI TEORIA

Il registratore è collegato al calcolatore tramite un bus (gruppo di piste elettriche) di sei linee, che fanno capo, sul retro del C/64, all'apposito connettore. Osservando quest'ultimo con attenzione sarebbe logico obiettare che, in realtà, vengano usate 12 linee, perchè il circuito stampato è a doppia faccia e presenta sei contatti sulla parte superiore e sei su quella inferiore.

In effetti, però, la faccia inferiore è una copia esatta di quella superiore, come è evidente osservando il manuale d'uso del C/64 (appedice I).

Le sei linee hanno varie funzioni: alimentazione, sensore di tasto premuto, motore on/off, lettura, scrittura e massa. Il segnale che la CPU invia al registratore in fase di scrittura è trasmesso immediatamente alla testina magnetica, senza che venga effettuata una trasformazione del segnale binario in frequenze standard rappresentanti lo 0 e l'1, (come, invece, accade nei modem).

Durante la fase di scrittura, sulla testina è sempre presente una polarizzazione magnetica, sia per il livello logico 1 che per quello 0. In modo simile, ma... opposto, avviene invece la lettura in cui il segnale del nastro, prelevato dalla testina, viene amplificato prima di essere rinviato alla CPU.

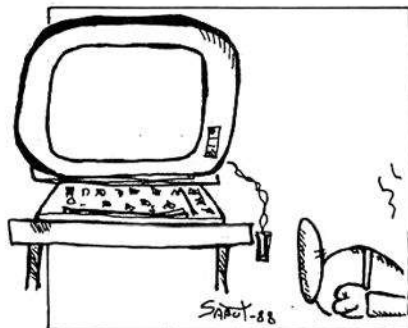
Questo lavoro è (fortunatamente) svolto in modo automatico dall'hardware, cosicchè al programmatore è sufficiente conoscere la posizione in memoria della porta cui è collegato il registratore perchè tutto funzioni a dovere.

Vale la pena di spendere due parole su questa porta, che si trova alle locazioni 0 e 1 e che, tra l'altro, comunica con un integrato apposito che si occupa di abilitare (o disabilitare) le ROM. Essendo questo integrato di importanza vitale per il corretto funzionamento del calcolatore, è meglio non manipolare le due locazioni se non si conosce perfettamente la loro funzione.

La porta in questione è, in realtà, composta da due registri. Il primo è quello di direzione dei dati, che si trova nella locazione zero, in cui 1 significa uscita e 0 ingresso per ognuno degli otto bit. Il secondo registro (il buffer dati) si incarica di riportare lo stato logico delle linee collegate. Modificando opportunamente il valore di queste due locazioni è possibile comandare a piacere il funzionamento del Datassette.

COME FUNZIONA

Chi desidera utilizzare il C/64 come un lettore di nastri musicali (evitando di approfondire i concetti teorici) deve limitarsi a seguire le note qui riportate:

- Digitare il programma Basic con la massima cura.
- Registrarlo e verificarlo (su disco o nastro).

IL MASSIMO DELLE POSSIBILITA'

La tecnica di base, descritta in queste pagine, non solo permette l'ascolto del nastro, ma, sofisticando la procedura, anche il campionamento dei suoni rilevati per poi effettuare varie elaborazioni, tra cui slittamenti in frequenza, salvataggi e/o riproduzioni degli stessi in propri listati.

Un programma più evoluto è stato infatti messo a punto (dallo stesso Lorenzo Emilitri) ed è presente nella cassetta "Software Club" che è possibile trovare in tutte le edicole.

• Inserire un joystick nella porta 2.
• Impartire il solito RUN.
• Inserire un nastro musicale nel datassette e regolare il volume del TV o del monitor.
• Per interrompere, premere Run/-Stop e Restore.
• Se il programma non si dovesse interrompere, muovere l'assicella del joystick.

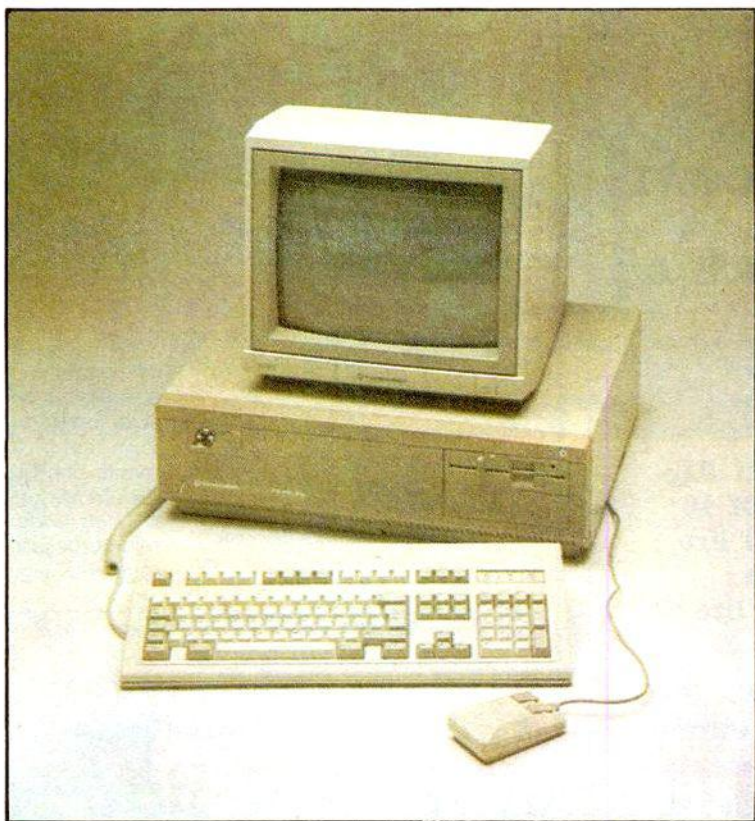
PERCHE' FUNZIONA

Il brevissimo programmino di queste pagine permette di ascoltare, attraverso l'altoparlante del TV o del monitor, il segnale presente su nastro (musica, voci, dati o altro). Come si può notare dal disassemblato, il listato risulta di facile comprensione. Per aumentarne la velocità operativa è stato utilizzato un trucco che si basa sulla gestione insolita della locazione 1.

La linea "lettura cassetta", infatti, è collegata al bit 3 della stessa, cosicchè qualsiasi suo mutamento cambia il valore del nibble basso di questa locazione. Dato che anche i registri $D020 (colore bordo) e $D418 (volume sonoro) usano il nibble basso per contenere, rispettivamente, il colore del bordo ed il volume, è sufficiente caricare l'accumulatore con il valore della locazione 1 e, successivamente, forzarlo in $D020 e $D418 per ottenere l'effetto voluto, vale a dire la visualizzazione ed emissione del segnale presente su nastro. Inoltre, come è chiaro dal disassemblato, è sufficiente muovere il joystick in porta due per ottenere il pronto ritorno al Basic.

```
100 REM TAPE LISTEN
110 REM BY L. EMILITRI
120 :
130 POKE 53280,0: POKE 53281,0
135 PRINTCHR$(147) CHR$(5)
140 FOR A = 49152 TO 49178
150 READ B: C = C+B
160 POKE A,B: NEXT A
170 IF C <> 3131THEN PRINT"ERRORE": END
180 PRINTTAB(10)"** ASCOLTA NASTRO **"
190 PRINT: PRINT"MUOVI IL JOYSTICK"
195 PRINT"IN PORTA #2 PER FINIRE"
200 SYS 49152
210 POKE 53280,0: END
220 REM DATA PER L/M
230 REM VEDI DISASSEMBLATO
240 DATA 120,169,000,133,000
250 DATA 173,000,220,133,002
260 DATA 165,001,141,032,208
270 DATA 141,024,212,165,002
280 DATA 205,000,220,240,241
290 DATA 088,096
```

```
        DISASSEMBLATO DEL PROGRAMMA
              "Ascolta Nastro"
            By Lorenzo Emilitri

; Definizione variabili:
;
JOY2   EQU $DC00 ;Joystck 2
BORDO  EQU $D020 ;Colore del bordo
VOLUME EQU $D418 ;Volume sonoro
TJOY2  EQU $02   ;Val. di confronto
                 ;con JOY2
;
       ORG $C000 ;Inizio: 49152
;
       SEI       ;Disab. interrupts
       LDA #$00  ;Abilita la porta I/O
       STA $00   ;come "ingresso"
       LDA JOY2  ;Salva il valore del
       STA TJOY2 ;joystick (2)
LOOP   LDA $01   ;Carica il valore
       STA BORDO ;e mettilo nel bordo
       STA VOLUME;e nel volume
       LDA TJOY2 ;Se e' stato mosso il
       CMP JOY2  ;joystick (2)
       BEQ LOOP  ;torna a LOOP
       CLI       ;Se no riabilita IRQ
       RTS       ;ed esce.
```

COMMODORE PC 60/40, IL PROFESSIONAL

Per pura curiosità, diamo uno sguardo alle caratteristiche di un tipo di elaboratore, basato sul micro 80386, che rappresenterà lo standard prossimo venturo

di **Antonio Pastorelli**

La Commodore, conosciuta ormai da tutti come azienda leader nella produzione di home computer, è da tempo entrata anche nel settore dei sistemi professionali, grazie alla proposta di macchine d'avanguardia.

Oltre ai compatibili PC-XT, come il PC-I, i PC10 e PC20, la parte del leone la fa sicuramente il PC 60/40, con un "cuore" a 32 bit effettivi (microprocessore 80386 Intel), che lavora alla velocità di 16 Mhz, contro gli 8 bit a 4.77 Mhz dell'8088 montato sulla maggior parte dei PC-XT Ms-dos compatibili.

La strategia di marketimg, infatti, si propone di soddisfare, con macchinari sempre più sofisticati, le esigenze dei nuovi utenti di prodotti informatici, tra cui si annoverano aziende medie e piccole e, spesso, anche privati.

La Commodore si è quindi introdotta nel mercato dei professionali proponendo un'ampia gamma di prodotti per promuovere l'evoluzione tecnica necessaria ad ogni impresa del settore; nonostante ciò, tuttavia, il mercato degli home computer non verrà abbandonato, ma, al contrario, sarà potenziato.

Vediamo ora quali sono le caratteristiche che fanno, del PC 60/40, un computer d'avanguardia.

La configurazione di base prevede tutto il necessario per lavorare in modo veloce ed efficiente:

- *1 Disco rigido da 40 Mbyte*

- *1 Drive da 5.25 pollici, ad alta capacità (1.2 Mbyte)*
- *Scheda grafica EGA WONDER*
- *2 porte seriali RS-232C*
- *2 porte parallele Centronics*
- *Memoria RAM da 2,5 Mbyte*

Dal momento che l'Ms-Dos non è in grado di gestire un hard-disk da 40 Mbyte, unitamente ai dischi di sistema viene fornita una utility che permette di creare partizioni, con le quali dividere i 40 Mbyte a disposizione in due (o più) unità logiche di memoria di massa.

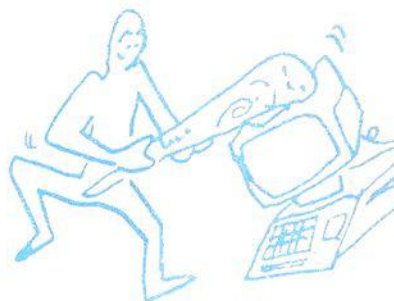
In tal modo si fa "credere", al sistema operativo, di avere a disposizione due (o più) hard-disk (ad esempio uno da 30 ed uno da 10 Mbyte), in modo che possa trattarli correttamente.

La scheda grafica a corredo si presta a qualsiasi esigenza: dal trattamento di testi alla gestione di grafici in altissima risoluzione.

La EGA WONDER, infatti, supplisce ad una varietà di numerose altre schede grafiche, attualmente in commercio, per Ms-Dos compatibili:

- *Color Graphics Adapter (CGA)*
- *Enhanced Graphics Adapter (EGA)*
- *Hercules Graphics Card (HGC)*
- *VGA*

Con la disponibilità dei quattro modi grafici, il computer è in grado di trattare testi dalle normali 80 fino alle 132 colonne, grafici dalla risoluzione minima di 320 x 200 alla massima di 800 x 560 con 16 colori scelti da una palette di 64 (monitor permettendo).

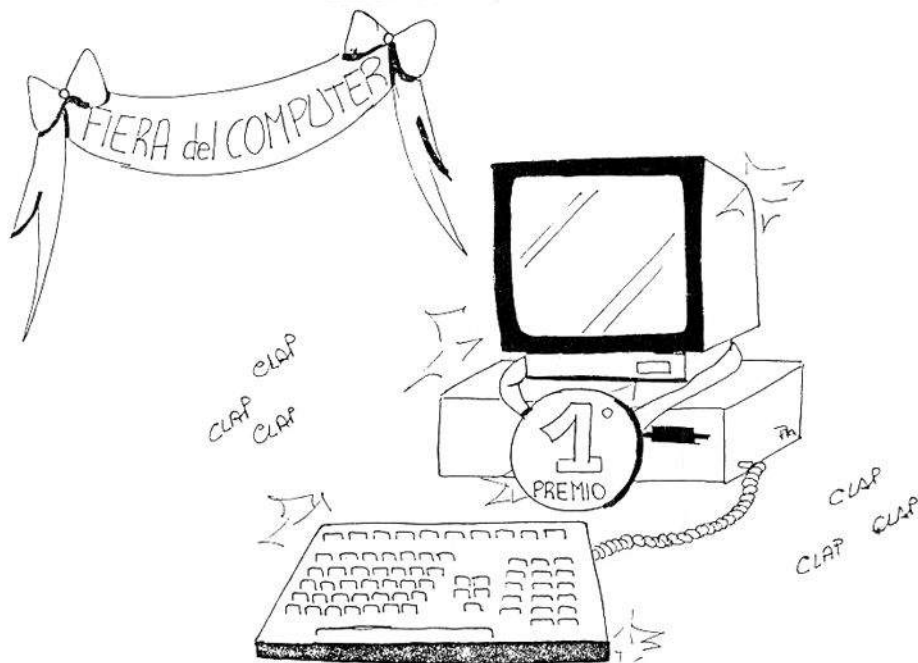
Il sistema, che si presenta con una linea compatta ed esteticamente ben curata, è fornito con un monitor monocromatico, chiave per la disabilitazione della tastiera, tastiera di tipo avanzato a 102 tasti, sistema operativo Ms-dos versione 3.21, Gw-basic 3.21; dispone di 8 slot, dei quali 6 liberi per eventuali espansioni.

Il computer, in seguito a prove, sia a livello hardware che software, risulta essere perfettamente compatibile con lo standard IBM, chiaramente con qualche nota di elogio dovuta alla velocità di elaborazione dei dati (punto di forza del potente microprocessore 80386), come dimostra la seguente tabella:

Test	Tempi PC 60/40	Tempi PC-IBM
Calcolo	2	16
Stringhe	0	2
Visualizzaz.	4	8
Scrittura file	14	93

Dal momento che l'80386 lavora ad una velocità decisamente superiore allo standard dei PC-IBM, per motivi di compatibilità è presente, nel disco di sistema, un comando che permette di selezionare la velocità di clock tra 8 e 16 Mhz.

Chi intende utilizzare più frequentemente la velocità più bassa, potrà inserire nella sequenza di bootstrap (l'insieme delle istruzioni che vengono eseguite per preparare il computer all'uso), il comando "Speed386 8", in modo da averla sempre selezionata all'accensione.

Il PC 60/40 si colloca dunque, di prepotenza, nella fascia dei computer professionali, spiccando per le sue prestazioni e per la completezza della dotazione di serie, che non trascura alcun tipo di destinazione.

Ciò viene confermato anche da recenti statistiche, che vedono la Commodore in vetta al mercato tedesco dei professionali (quelli dalle 750 mila ai 13 milioni), con una quota pari al 17,4%, mentre sul mercato europeo si colloca in un dignitoso terzo posto.

COMMODORE A TUTTO VIDEO (PER NON PARLAR DELL'AUDIO)

Un gruppo di lettere giunte in redazione sono un'ottima scusa per parlare di un argomento interessante

di **Giancarlo Mariani**

Il Commodore 64, ma soprattutto il C/128, è fornito di varie uscite video che vanno (per quest'ultimo) dalla monocromatica alle 80 colonne a colori RGB. A tale varietà di modi di visualizzazione corrisponde, sul C/128, un certo numero di connettori per collegare l'apposito monitor in modo giusto.

Lo schema e le funzioni dei connettori sono riportate nel manuale, ma da alcune lettere che pervengono in redazione sembra che qualche lettore abbia le idee poco chiare in merito.

Tratteremo, quindi, alcuni dei problemi riguardanti le uscite audio/video del C64/128, emersi dalle lettere giunte.

Il primo lettore è Riccardo Maffini (di Mede), che ha le classiche "poche idee ma confuse" sulla sistemazione dei piedini del connettore audio/video del C/64. La corretta sistemazione di questi piedini è mostrata in figura 1. Un appunto da fare è che sul manuale i piedini 7 e 8 sono segnati come NC (No Connection = nessuna connessione), ma su alcuni esemplari di C/64/128 possono portare una tensione di +5 Volt oppure altre... "cose", tra l'altro facilmente individuabili mediante un comune tester.

Glauco Bruno, di Milano, ha dubbi sul corretto collegamento tra il C/64 ed un videoregistratore perchè dichiara di ottenere solo "immagini decenti ma non perfette". Abbiamo effettuato anche noi il collegamento ma, ad onor del vero, i risultati ottenuti li abbiamo giudicati piuttosto buoni anche se l'immagine riprodotta non è "pulita" come quella visualizzata direttamente dal computer.

I collegamenti effettuati sono mostrati in figura 2 (ci riferiamo ad un VCR dotato di presa Scart). Il segnale video del computer viene prelevato tramite i piedini 2 (GND) e 4 (Composite Video Out) del connettore del computer, mentre l'audio si preleva dai piedini 2 (che è ancora la massa) e 3 (audio out). Gli ingressi sulla presa Scart, mostrata nella stessa figura 2, sono i piedini 4 e 6 (audio) ed i piedini 17 e 20 (video).

Non bisogna dimenticare di collegare la schermatura della Scart (piedino 21 o contenitore metallico) con

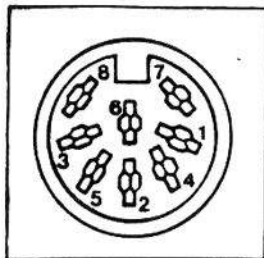
Connettore audio / video C/64/128

Pin	Type	Note
1	LUM/SYNC	Luminance/SYNC output
2	GND	
3	AUDIO OUT	
4	VIDEO OUT	Composite signal output
5	AUDIO IN	
6	COLOR OUT	Chroma signal output
7	NC	No connection
8	NC	No connection

Fig. 1

la custodia metallica del connettore video del computer. I collegamenti tra VCR e Televisore potranno essere effettuati allo stesso modo con cui si opera per le normali videocassette, cioè tramite uscita video (soluzione ottimale) o tramite presa antenna TV.

Ed ora è la volta di Franco Di Marino (di Martina Franca) che sottopone alcune domande alle quali cercheremo di dare risposte sufficientemente chiare.

La prima riguarda la massa da utilizzare per il collegamento in 80 colonne con un normale televisore (vedi C.C.C. n. 29, Marzo 1986). Abbiamo provato il collegamento, descritto nell'articolo, che funziona correttamente; è probabile, tuttavia, che vi siano imprecisioni circa il connettore RGB del C/128. In ogni caso riportiamo, in queste stesse pagine, la corretta disposizione dei piedini del connettore, che è stato accuratamente verificato dopo numerosi tentativi, anche a rischio e pericolo del nostro povero computer. Tale disposizione è mostrata in figura 3.

Il secondo dubbio richiede chiarimenti riguardo il piedino 5 del con-

Fig. 2

nettore Audio / Video del C/64, siglato "Audio In" (Ingresso Audio).

Si vocifera da tempo che, tramite questo piedino, sia possibile inviare un qualsiasi segnale sonoro al calcolatore, onde manipolarlo (il segnale, non il computer) tramite filtri e registri disponibili nel SID ed infine porlo nuovamente in uscita. Purtroppo non abbiamo avuto modo di realizzare un circuito elettronico (o un collegamento) che sia in grado di effettuare quanto appena descritto, ma se qualche lettore dovesse saperne qualcosa... bè: siamo qui per pubblicare le notizie relative.

L'ultimo dubbio riguarda la visualizzazione delle 80 colonne su di un televisore dotato di presa Scart, problema, a quanto pare, che ha deluso molti possessori di C/128.

Pur non avendo avuto l'opportunità di soffermarci sul caso (siamo tutti fortunati possessori di monitor specifici...) riteniamo che, forse, vi sia un problema di incompatibilità tra i livelli di uscita del 128 e quelli di ingresso del TV. I primi, infatti, sono livelli TTL (segnale RGB digitale, ossia i livelli possono essere solo 0 oppure 5V), con addirittura un piedino che controlla l'intensità del colore; i secondi, invece, hanno quasi sempre ingressi di tipo analogico, con tensione variabile.

Ed ora è la volta di Sandro Enea (di Palermo), che possiede un monitor Commodore monocromatico (vedi CCC n. 48 pag. 66) e non riesce a visualizzare le 40 colonne del C/128. Innanzitutto chiariamo che non sembra possibile che dalla porta del Joystick n.1 esca un segnale video, come riferito dal commesso di un negozio situato nella città del nostro lettore.

Crediamo, piuttosto, che il problema riguardi un'incompatibilità di livelli tra il C/64 ed il monitor, dal momento che questa volta (al contrario

Connettore RGB C/128

Pin	Signal
1	Ground
2	Ground
3	Red
4	Green
5	Blue
6	Intensity
7	Monochrome
8	Horizontal Sync
9	Vertical Sync

Fig. 3

del caso precedente) è il monitor ad avere ingressi TTL, mentre l'uscita a 40 colonne del C/128 è di tipo analogico. Un altro problema potrebbe riguardare il fatto che il monitor possiede due piedini di ingresso per la sincronizzazione orizzontale e verticale, cose che invece sono già presenti nell'unico piedino di uscita a 40 colonne del computer. Non avendo sottomano il monitor incriminato, non possiamo essere di aiuto; si può, comunque, provare ad effettuare il collegamento di figura 4. Se funziona... bene, altrimenti...

In ogni caso, anche se non dovesse funzionare, non si rischia di danneggiare nè il monitor nè il computer.

Il piedino intensità (n. 4 del connettore del monitor), è bene PRIMA collegarlo a massa; solo in seguito, se non dovesse funzionare, alla tensione +5 Volt. Ricordiamo che i +5 Volt sono presenti sul piedino B-2 della porta del registratore del C/128.

Infine passiamo a Piercarlo Saracco (di Torino), che resterà forse deluso dal momento che, per quanto ne sappiamo, non esiste in commercio una scheda per trasformare un segnale video composito (uscita a 40 colonne del 64/128), in un segnale RGB, da inviare al monitor a colori posseduto dal lettore.

Audio/Video

Pin	Tipo	Nota
1	LUMINANZA	
2	MASSA	
3	USCITA AUDIO	
4	USCITA VIDEO	
5	INGRESSO AUDIO	

C/64/128

7AV106

MONITOR

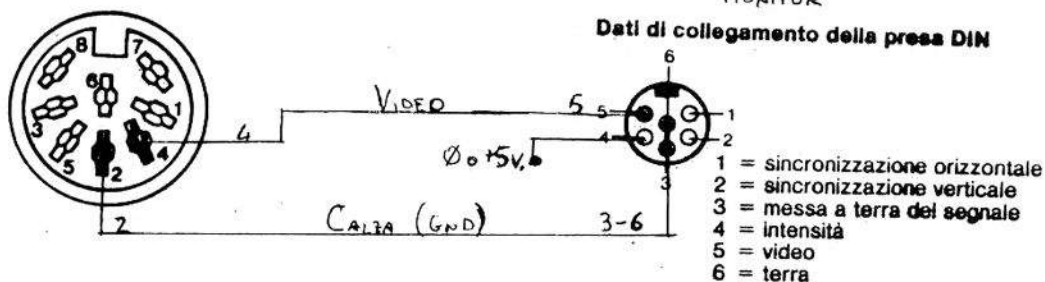
Dati di collegamento della presa DIN

VIDEO

Ø o +5V

CALZA (GND)

3-6

1 = sincronizzazione orizzontale
2 = sincronizzazione verticale
3 = messa a terra del segnale
4 = intensità
5 = video
6 = terra

Cassette

Pin	Type
A-1	GND
B-2	+5V
C-3	CASSETTE MOTOR
D-4	CASSETTE READ
E-5	CASSETTE WRITE
F-6	CASSETTE SENSE

1 2 3 4 5 6

A B C D E F

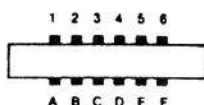
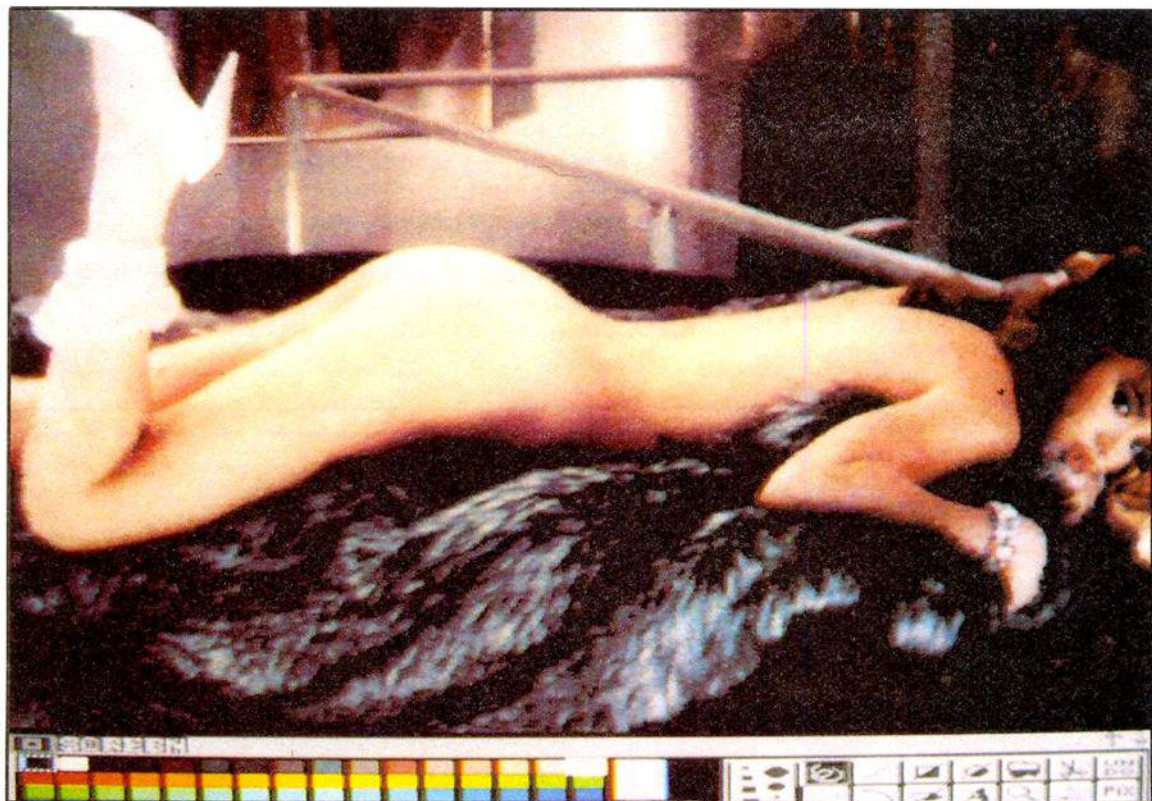



Fig. 4

SIETE PREGATI DI ALLACCIARE LE STRINGHE

Il linguaggio C prevede funzioni molto elastiche per gestire parole e frasi

di **Luigi Callegari**

Il linguaggio C, come già detto in precedenza, non possiede il "tipo stringa" di caratteri tra i propri tipi base, ma solo il tipo "carattere" (char). Perciò una sequenza di più caratteri deve essere dichiarata come una matrice (array) di caratteri singoli, come, ad esempio:

char stringa[24];

In questo caso, "stringa" è un array di 24 caratteri nel senso, però, che potremo memorizzare al massimo 23 caratteri! Infatti il C, per convenzione, chiude sempre una stringa con un carattere di NULL, owero con un byte pari a zero, owerossia simbolicamente col carattere backslask e zero (0). Ad esempio...

char *s = "Gabry"

...sarà fisicamente ed automaticamente memorizzato con seguenti byte...

47 61 62 72 79 00

...owero con i codici ASCII (esadecimali, vedi appendice A-2 del manuale AmigaBasic) delle cinque lettere oltre al carattere NULL. Il compilatore memorizza la stringa da

qualche parte (in Amiga, nel cosiddetto HUNK CODE, ovvero nella sezione di codice riservata alle costanti) ed assegna alla variabile "s" un puntatore al primo byte (47h), ovvero il suo indirizzo in memoria.

E' essenziale ricordarsi di questa convenzione del linguaggio C: le stringhe devono essere sempre "NULL-terminated", onde permettere al programma di rilevarne dinamicamente la lunghezza (quando si legge uno zero vuol dire che la stringa è finita). Si noti che ciò permette una gestione molto flessibile e compatta delle stringhe in linguaggio C in quanto una stringa (parola o frase) può essere gestita interamente tramite un solo parametro: il suo indirizzo contenuto in un'appropriata variabile puntatrice (char *nome).

Tale convenzione permette di scrivere funzioni di manipolazione delle stringhe alfanumeriche, anche sofisticate, con pochi comandi.

```
#include "stdio.h"
main()
{
char stringa[12];
int  lungh;
printf("Batti linea ");
fgets(stringa,11,stdin);
lungh=strlen(stringa)-1;
printf(" \nLunga: %d \n",lungh);
}
```

Figura 1

MEMORIZZARE PAROLE

Esaminiamo ora il banale programmino di figura 1, che richiede una stringa e ne stampa la lunghezza, commentandolo linea per linea:
1) Si include il file di inclusione "stdio.h" perchè intendiamo utilizzare funzioni e parametri standard di I/O (printf(), fgets(), stdin).
2) Vi è una sola funzione, quella principale: main().
3) Dovremo usare una stringa lunga al massimo 11 caratteri, perciò dichiariamo una matrice (array) di 12 caratteri.
4) Richiediamo la stringa con un normale printf() che scrive, per default, sul video.
5) Usiamo la funzione fgets() per leggere la stringa. Essa prevede tre parametri: un puntatore all'array, dove memorizzare i caratteri che arriveranno (stringa); il numero massimo di caratteri da accettare (11) ed il dispositivo da cui leggere i caratteri. Nel nostro caso specifichiamo l'input standard (stdin), ovvero la tastie-

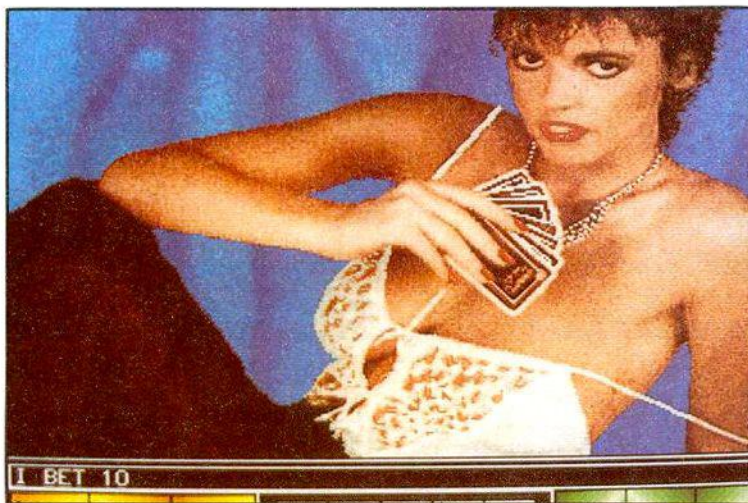
NON SOLO NUMERI

Per un elaboratore elettronico è fondamentale gestire le cosiddette stringhe alfanumeriche, ovverossia caratteri, simboli e lettere, oltrechè numeri. Buona parte dei programmi commerciali, infatti, opera più con parole che con aridi numeri; basti pensare alle videoscritture oppure ai database.

Per i personal computer, come i nostri C/64 e Amiga, probabilmente le parole sono ancora più importanti in quanto adatte ad uso "domestico" e non usate, di solito, per sofisticati calcoli matematici. Molto probabilmente chi scrive programmi in C dovrà dunque sapere molto bene come gestire lettere, parole e frasi.

```
#include "string.h"
size = stccpy(to,from,n);
np   = stpcpy(to,from);
p    = strcpy(to,from);
p    = strncpy(to,form,n);

char *np; fine stringa destinazione
char  *p; identico alla destinazione
char *to; puntatore di destinazione
char *from; puntatore sorgente
int n; massima lunghezza sorgente

Figura 2
```

ra, ma si potrebbe anche specificare un file.

6) Calcoliamo con la funzione strlen() la lunghezza della stringa (simile alla funzione LEN del Basic). Il parametro è un puntatore a carattere (stringa).

Sono da notare molte cose, in queste poche linee.

Innanzitutto si usa il nome della matrice di caratteri come parametro delle funzioni, che attendono come parametro un puntatore a carattere (come ricorda il manuale d'uso del compilatore). I nostri lettori ricorderanno certamente che in C il nome di una matrice viene risolto dal compilatore come se fosse un puntatore al primo carattere dell'array.

Si noti anche la dichiarazione di stringa con dodici elementi, anche se, accessibili, saranno solo 11, perchè in fondo dovremo considerare il carattere NULL.

Inoltre l'uso di 11 come numero massimo di caratteri in fgets(), che altrimenti riempirebbe tutti i dodici posti riservati a lettere con i codici ASCII che gli arrivano, senza potere poi chiudere col NULL la stringa (cosa che potrebbe provocare poi una visita del GURU!).

Inoltre la lunghezza effettiva della stringa viene diminuita di uno, in quanto la funzione fgets() legge e memorizza anche la pressione del tasto Return passando il codice ASCII 13 e memorizzandolo in "stringa" come un qualsiasi altro carattere.

FUNZIONI DI STRINGA

Oltre alla nota strlen(), di cui ci siamo occupati nei fascicoli precedenti, esistono molte altre funzioni nella libreria standard del C.

Ricordiamo strcpy(), strcmp(), stcarg(), strcat(), stcmp(), stpsym(). Comunque, è bene consultare la descrizione del manuale del vostro compilatore per sapere di che cosa disponete in libreria; qui facciamo riferimento a funzioni basilari pre-

ALLA RICERCA DEGLI ESEMPI

Siamo fermamente convinti, per esperienza personale, che non si può imparare un linguaggio di programmazione senza avere a disposizione un gran numero di esempi. Purtroppo il linguaggio C è abbastanza nuovo nel campo dei computer personali, perciò gli esempi di listati da studiare scarseggia. Se poi si desiderano programmi semplici, magari commentati in lingua italiana, gli esempi sono pressochè inesistenti.

Per questo motivo abbiamo pensato di inserire regolarmente nei nostri dischetti mensili "AmiGazzetta", numerosi esempi di programmi scritti in C, commentati quasi sempre in italiano. Si possono trovare, sui dischetti usciti sino ad ora, sia programmi semplici (Crypt, Gioco dell'indovina parola) sia programmi di media complessità (compressore di file). Tutti i programmi sono comunque istruttivi e, ciò che più conta, funzionanti!

senti sia in "Lattice Amigados C" che in "Super C-64".

Alcune di queste funzioni hanno molte sorelle quasi omonime. Ad esempio, cercando nei pressi di strcpy() si trovano anche: stccpy(), stpcpy(), strncpy(). Ciò almeno nel manuale Lattice. Queste funzioni svolgono in realtà sempre compiti molto simili, ma con sfumature di differenza. Inoltre talune sono inserite per compatibilità "incrociate" con altri compilatori non proprio standard, ma sono identiche ad altre. Vediamone qualcuna.

COPIA DI STRINGHE

Le funzioni di ricopiatura di stringhe suonano nuove ai programmatori Basic abituati a scrivere, ad esempio,...

```
a$ = b$
c$ = "Gabriella"
```

...per copiare stringhe. Però il Basic possiede la stringa tra i suoi tipi base, mentre in C è un tipo costruito, come abbiamo detto, assemblando in matrici dei caratteri; servono quindi apposite funzioni per manipolarle. Le relative sintassi formali sono riportate in figura 2.Tutte queste funzioni copiano materialmente una stringa sorgente (from) in una stringa destinazione (to). La stringa sorgente deve terminare con un NULL.

Nel caso di strcpy() e stpcpy() viene ricopiata la stringa sorgente intera e la stringa di destinazione viene automaicamente chiusa con un NULL (byte a zero).

Nel caso si strncpy(), vengono sempre scritti "n" caratteri nella destinazione. Se la sorgente è più corta, lo spazio della stringa di destinazione viene completata con NULL; se invece la stringa sorgente è più corta di "n" verranno copiati comunquue "n" caratteri, ma mancherà il NULL di chiusura, con le conseguenze del caso.

Stccpy(), invece, è simile a strncpy() ma produce sempre una stringa chiusa (terminante con NULL) e restituisce il numero effettivo di byte posti nell'area di destinazione.

Si noti che la stringa di destinazione dovrà sempre essere stata dichiarata come una matrice di caratteri capace di contenere TUTTI i caratteri che le funzioni di copiatura le invieranno, altrimenti avremo a che fare col GURU durante l'esecuzione del nostro programma.

Riportiamo, in figura 3, una delle

```
void strcpy(a,b)
char *a,*b;
{
/* Copia a in b */
  while (*a++=*b++)
        ;
}
```

Figura 3

```
#include <string.h>
l = strlen(s);
l = stclen(s);
int l; /* Lunghezza */
char *s; /* Stringa */
```

Figura 4

```
int x,y;
char *s="Picchio";
x=strlen("old");
y=strlen(s);
```

Figura 5

```
#include <string.h>
x=strcmp(a,b);
x=strcmpi(a,b);
x=stricmp(a,b);
x=strncmp(a,b,n);
x=strnicmp(a,b,n);
int x;
int n;
char *a,*b;
```

Figura 6

```
strcmp(a,b)
char *a,*b;
{
for ( ; *s=*t ; s++,t++)
  if (!(*s) )
      return(0);
return(*s -  *t);
}
```

Figura 7

EVITATE LE MEDITAZIONI DEL GURU

Quando si manovrano con il C di Amiga le stringhe di caratteri, al momento dell'esecuzione si incontra facilmente il Guru. Questo scoraggia molti principianti, abituati dal Basic a correggere rapidamente ogni errore e a rilanciare subito l'esecuzione.

In C, invece, è sufficiente copiare una stringa in un'altra, dimensionata troppo corta con un strcpy(), senza avere segnalazioni di errori dal compilatore; questo, infatti, non può prevedere che una stringa sia abbastanza ampia per ciò che vorremo metterci, ma blocca inesorabilmente la macchina al momento dell'esecuzione. In questo caso, il rimedio non è correggere e ridare RUN, ma spegnere, riaccendere, ricaricare l'editor, correggere, ri-compilare, ri-linkare e rilanciare!

Ciò si riallaccia a quanto affermato nei fascicoli precedenti: il C, essendo un linguaggio COMPILATO e non INTERPRETATO, non permette scritture di programmi a "correzioni d'errori", ma impone di studiare sempre a tavolino il problema ed a lavorare quanto più possibile con carta, matita e meningi prima di picchiare sui tasti.

stesure possibili relative ad una funzione-tipo strcpy().

STRLEN()

Esistono in C due funzioni identiche, ma dotate di nomi diversi, che restituiscono il numero di caratteri (ad eccezione di Null) presenti in una stringa. I formati sono riportati in figura 4.

Ad esempio, le linee indicate in figura 5 assegnano, alle variabili x ed y i valori 3 e 7.

STRCMP()

Come detto più volte, non è possibile, in C, confrontare direttamente due stringhe. Gli operatori di confronto (= =, >=, != ...) si possono applicare tranquillamente a numeri interi, in virgola mobile ed a singoli caratteri, ma non a stringhe. Perciò si deve usare una funzione di libreria da definire sempre nel file di testata "string.h".

Questa, presi due puntatori a stringhe ("a" e "b"), restituirà un valore negativo se la prima stringa è minore della seconda, un valore positivo se la prima è maggiore della seconda ed un valore pari a zero se sono identiche.

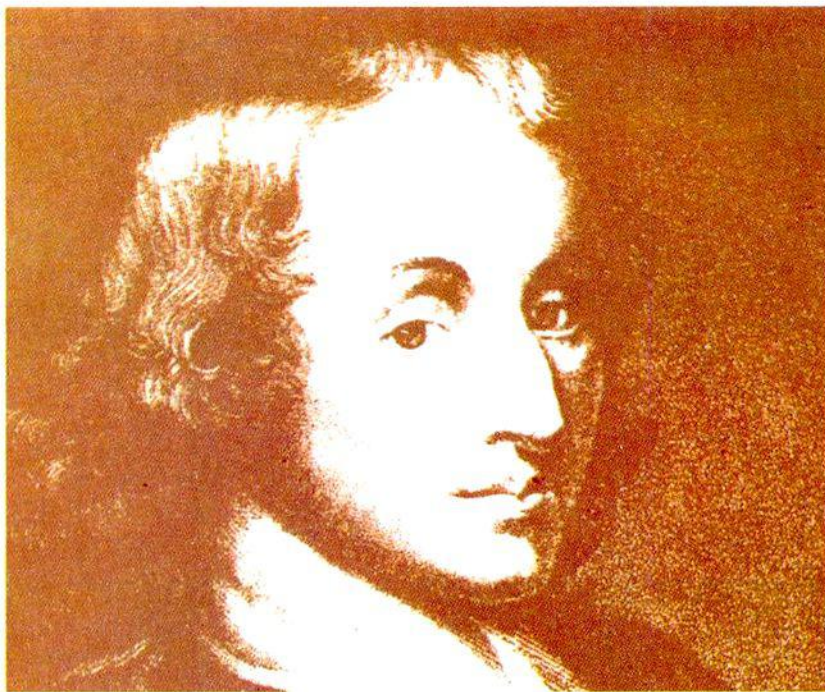
Le funzioni di confronto di stringa, disponibili in Lattice C, sono riportate in figura 6.

Tutte queste funzioni confrontano i valori numerici ASCII di due stringhe (a, b) terminate da NULL e forniscono un valore appropriato (positivo o negativo) quando incontrano due valori diversi, ovvero il numero zero se raggiungono la chiusura di ambedue senza rilevare differenze.

Le funzioni strcmpi(), stricmmp() e stricmmp() non "vedono" distinzioni tra caratteri maiuscoli e minuscoli durante il confronto. La funzione strcmpi() è identica a stricmp(); la prima è presente in vecchie versioni Microsoft, mentre la seconda è standard ANSI.

Nel caso di strncmp() e strnicmp() vengono confrontati al massimo "n" caratteri.

In figura 7 è riportato un esempio didattico di come si potrebbe scrivere strcmp().

Altre funzioni di manipolazioni di stringhe saranno esaminate in dettaglio prossimamente, oppure riportate sul dischetto mensile "AmiGazzetta" che viene venduto, dalla Systems Editoriale, solo per corrispondenza.

BORLAND TURBO PASCAL PER C/128

Una fantastica opportunità per chi ha deciso di usare in modo serio l'altro computer celato nel C/128

di **Alessandro de Simone**

Chi ha acquistato il C/128 dotandolo di disk drive 1541 (meglio se 1571) è certamente rimasto deluso per la quasi totale carenza di software specifico sia per la modalità 128 che per quella CP/M.

Da qualche tempo, però, la prestigiosa software house americana "Borland International" (la cui sede italiana è in Viale Cirene 11, Milano - tel. 02-588.523) ha deciso di saggiare il mercato di questo piccolo computer proponendo una valida versione di "Turbo Pascal" specifica per il C/128.

La Borland, infatti, è nota ai più grazie ai package destinati ai computer professionali che utilizzano, in special modo, il sistema operativo Ms-Dos. Tra i suoi prodotti figurano vari linguaggi (tra cui il C, il Turbo Pascal, varie versioni di Basic e di compilatori) e numerosi pakage di alto livello tra cui spreadsheet (lavagne elettroniche), database (per archivi), word processor e così via.

In precedenza la Borland aveva divulgato una versione per CP/M e, di conseguenza, l'adattabilità al C/128 è risultata piuttosto semplice.

LA CONFEZIONE

La confezione è costituita dal voluminoso manuale (in italiano!) di quasi 400 pagine e da un dischetto, sigillato.

Abbiamo provato ad usare il floppy sia con il C/128-D che con un C/128 collegato ad un drive 1541 compatibile (ed esattamente il modello RF 501-C, messoci gentilmente a disposizione dalla ditta Circe di Milano, viale F. Testi 219, tel. 02/64.27.410).

In entrambi i casi il compilatore ha

I TRE COMPUTER DEL C/128

Il Commodore 128 è venduto, ormai, sono nella versione C/128-D dotata, cioè, di disk drive 1571 al suo interno.

La produzione dell'altro modello (il C/128 "semplice") è stata ritenuta antieconomica e al di fuori dei tempi.

Un computer "serio" della categoria cui appartiene il C/128, infatti, non può fare a meno del drive; se ne sono accorti gli utilizzatori che, negli ultimi tempi, hanno trascurato di acquistare il modello base costringendo la Commodore a porlo fuori produzione.

Il C/128-D è un ottimo apparecchio, relativamente economico (solo 895000 lire) e contiene, in realtà, ben tre computer: il C/64, il C/128 (cioè un C/64 dotato di maggior quantità di memoria e di un Basic esteso) in modalità, tra l'altro, 40 oppure 80 colonne, ed il CP/M anch'esso gestibile in 40 oppure 80 colonne.

Il CP/M è l'obsoleto sistema operativo basato sul microprocessore Z-80; tuttavia, nonostante l'età, il CP/M dimostra di offrire, ad un grosso pubblico, software professionale di un certo pregio.

Il formato dei dischetti necessario per operare con il C/128 ha impedito, nella stragrande maggioranza dei casi, di entrare in possesso della miriade di programmi disponibili per tale valido sistema operativo.

Pur se tardivamente, però, il lavoro sotterraneo di molti appassionati, che operano soprattutto a livello di scambi, ha consentito di considerare con nuovi occhi le potenzialità finora trascurate nel C/128.

funzionato egregiamente; naturalmente con il 501 (come pure con un 1541 normale) la velocità è decisamente minore se confrontata con il 1571 contenuto nel C/128-D, pur se sopportabile.

Il dischetto non è protetto in alcun modo ed è possibile effettuare copie di sicurezza servendosi della solita procedura PIP disponibile sul dischetto originale del C/128. E' noto, tuttavia, che non serve a nulla disporre del software (soprattutto se questo è un linguaggio!) se non si dispone anche del manuale relativo.

Il prezzo al pubblico (L. 179 mila + I.V.A.) sembra dunque contenuto, soprattutto tenendo conto della qualità del software offerto.

POTENZIALITA'

Dando uno sguardo al manuale, ed esaminando le videate sullo schermo, possiamo affermare che la versione per questo piccolo computer è praticamente identica a quella che gira sui calcolatori Ms/Dos, velocità di elaborazione a parte (non bisogna dimenticare che si dispone di un semplice Z-80 ad otto bit!).

Se, poi, si ha la possibilità di usare il monitor in modo 80 colonne, la somiglianza aumenta ed i risultati sono di tutto rispetto.

Naturalmente sono assenti le istruzioni grafiche, introdotte nelle recenti versioni Ms-Dos, ed alcune utility più "moderne". Ai tempi del CP/M, infatti, alcune opzioni erano un vero lusso o risultavano addirittura impossibili. La sintassi di base, tuttavia, è rigorosamente identica alle versioni più nuove e perfino l'editing, sofisticato come un word procesor, non lascia nulla a desiderare.

La versione Turbo Pascal 3.0 per C/128, di cui ci stiamo occupando, offre le sei classiche opzioni:

- Edit, per scrivere il programma che, in seguito, verrà compilato in linguaggio macchina.
- Compile, appunto, per effettuare l'operazione di compilazione; è possibile attivare, o meno, l'emissione dei messaggi di errore che, in caso contrario, vengono visualizzati con il solo numero di codice.
- Run, come intuitivo, serve per far "girare" il programma compilato.
- Save, Directory e Quit, infine, permettono, rispettivamente, di registrare su disco il programma (sia il sorgente che il compilato), di esami-

"COMMODORE COMPUTER CLUB" AFFRONTA IL TURBO PASCAL

Come promesso tempo fa, avvisiamo i nostri lettori che verranno presto divulgati articoli relativi a programmi scritti in Turbo Pascal.

Naturalmente verranno privilegiati gli articoli relativi a programmi (essenzialmente brevi e didattici) che possano girare sul C/128 in modo CP/M ma, soprattutto, utili per varie applicazioni.

Si ricorda che qualsiasi programma, che possa girare in modo CP/M, girerà anche in modo Ms-Dos, disponendo, ovviamente, di un computer IBM compatibile e di linguaggio Turbo Pascal per Ms-Dos.

Coloro che intendessero collaborare in tal senso possono contattare la redazione per proporre gli argomenti da affrontare.

```
A>dir
A: CPM+     SYS : CCP      COM : HELP     COM : HELP     HLP : KEYFIG   COM
A: KEYFIG   HLP : FORMAT   COM : PIP      COM : DIR      COM : COPYSYS  COM

A>dir
A: DATE     COM : DATEC    ASM : DATEC    RSX : DEVICE   COM : DIR      COM
A: DIRLBL   RSX : DUMP     COM : ED       COM : ERASE    COM : GENCOM   COM
A: GET      COM : INITDIR  COM : PATCH    COM : PIP      COM : PUT      COM
A: RENAME   COM : SAVE     COM : SET      COM : SETDEF   COM : SHOW     COM
A: SUBMIT   COM : TYPE     COM : SETUP    COM
```

Directory del dischetto CP/M originale Commodore

```
A>dir
A: TURBO    OVR : TURBO    MSG : TINST    COM : READ     ME  : TINST    MSG
A: LISTER   PAS : CMDLIN   PAS : MC       PAS : MC-MOD00 INC : MC-MOD01 INC
A: MC-MOD02 INC : MC-MOD03 INC : MC-MOD04 INC : MC-MOD05 INC : MC       HLP
A: MCDEMO   MCS : TINST    DIA : TURBO    COM
```

Directory del dischetto originale Borland Turbo Pascal 3.0

nare la directory (e di cambiar numero di device; a proposito, abbiamo constatato che operando con due drive, n. 8 e 9, tutto funziona regolarmente) e di abbandonare il Turbo Pascal ritornando al tipico prompt del CP/M.

Per ciò che riguarda il linguaggio, saremo succinti: sono sfruttabili le variabili integer, boolean, real, char; le etichette, le procedure e le funzioni; gli operatori sono quelli ben noti e tra le istruzioni figurano, ovviamente, quelle strutturate le iterative. Non mancano gli array e le operazioni relative nè i record, i set e le numerose procedure che sono alla base di questo moderno ed attuale linguaggio compilatore.

Nulla, insomma, è stato trascurato per offrire all'utente del C/128 (purchè dotato di disk drive) uno strumento software valido per accedere al mondo degli elaboratori professionali.

COPIARE E' FACILE

Operando con il C/128 in modalità CP/M, e con un drive 1541 (o compatibile) ci si accorge che i tempi di lettura e scrittura su floppy sono spesso rilevanti. La tabella di queste pagine riproduce il confronto riscontrato tra il 1571 contenuto nel C/128-D ed un drive 1541 compatibile.

Pur se, in effetti, i tempi sembrano più che sopportabili, l'utente del C/64, abituato ai vari tool velocizzatori in circolazione, li può giudicare eccessivi.

Per evitare inutili perdite di tempo, ma soprattutto per evitare cancellazioni accidentali, consigliamo di effettuare sempre una copia di sicurezza del software, prima di usarlo. Questo consiglio, del resto, è caldeggiato in qualsiasi manuale che si rispetti.

I file presenti sul disco Turbo Pascal 3.0 sono numerosi e non tutti risultano indispensabili per scrivere programmi.

Si consiglia, pertanto, di formattare un dischetto vergine (in modo, ovviamente, CP/M) e riportarvi tutti i file che si intendono utilizzare.

Si tenga presente che, diponendo di un 1541 (o compatibile), può risultare necessario copiare soltanto pochi file, trascurando accuratamente di riportare, sul disco copia, i file che non occorrono; non dimentichiamo, infatti, che i 160 Kbyte si esauriscono rapidamente, soprattutto tentando di registrare, sullo stesso dischetto, più versioni successive di uno stesso programma (sorgente, compilato eccetera).

FUNZIONE SVOLTA	TEMPO (in secondi) 1571	1541
Boot CP/M	21	120
Caricamento comando Format	3	20
Esecuzione comando Format		87
Caricam. e attivazione Turbo Pascal	26	100
Caricam. dei messaggi del Turbo Pascal	3	8

Tempi necessari per eseguire alcune funzioni CP/M con il C/128 collegato ad un drive 1571 oppure 1541

Consigliamo, quindi, di realizzare due dischi: il primo contenente i file del sistema CP/M di uso più frequente (PIP, FORMAT e così via) ed un secondo floppy sul quale saranno presenti esclusivamente i file vitali del Turbo Pascal. All'occorrenza, in seguito, si effettueranno altre copie, oppure si provvederà a cancellare i file che non interessano.

Ricordiamo, comunque, che sul manuale del C/128 (in italiano) l'intero quarto capitolo è dedicato alla gestione del computer nella modalità CP/M.

Per comodità dei pigri, tuttavia, riportiamo la procedura necessaria per copiare, su un disco vergine, l'intero contenuto di un dischetto CP/M, pur disponendo di un solo drive:

- Introdurre copia del dischetto CP/M originale Commodore (lavorate sempre con le copie, MAI con gli originali!).
- Digitare Format e premere Return.
- Alla richiesta, estrarre il dischetto copia, introdurre quello vergine e rispondere alle domande che verranno poste.
- Reintrodurre il disco sistema CP/M e digitare PIP
- All'apparire del prompt specifico di PIP (cioè l'asterisco: *), estrarre il disco sitema, introdurre il disco da copiare e digitare...

e:=a:*.*

...che significa: copia nel disco "e" (virtuale) tutti i file che sono presenti nel drive al momento in cui si preme il tasto Return.

A questo punto è sufficiente estrarre il floppy originale od introdurre la copia, a seconda dei messaggi che, a mano a mano, compaiono sul video, fino ad ultimare le operazioni.

CAMPUS

LA TUA UNIVERSITA' DEL SOFTWARE

LABORATORIO SOFTWARE DI COMMODORE COMPUTER CLUB

ESORDIENTI

- Sbagliando s'impara

ESPERTI

- Spremi il cervello del tuo C64
- Tutte le SYS del 64

CAMPIONI

- Sfruttiamo il S.O. del C128

La Grande Libreria Systems

Autori Vari

64 Programmi per Commodore 64

Giochi, grafica, gestione delle stringhe, musica, numeri, gestionali.

Lire 4.800

Autori Vari

I miei amici C16 & Plus4

Un manuale pratico per padroneggiare il basic di questi computer.

Lire 7.000

Autori Vari

Strategie vincenti per Commodore 64

Le strategie per tutti i classici del videogioco: per giocarli, vincerli o programmarli.

Lire 5.800

Autori Vari

62 Programmi per il Vic 20, C16 e Plus 4

Giochi, grafica e routine per imparare a programmare.

6.500

Roberto Didoni, Guido Grassi

Utilities e giochi didattici

Raccolta di programmi pratici per tutti i Commodore e lo Spectrum.

Lire 6.500

Giovanni Mellina

Tutti i segreti dello Spectrum

4 passi nella Rom: come usare le più importanti routine del sistema operativo.

Lire 7.000

Roberto Didoni, Guido Grassi

Simulazioni e test per la didattica

Teoria e listati per Vic 20, C16, C64 C128 e Spectrum Sinclair.

Lire 7.000

Paolo Goglio

Impara giocando il basic dello Spectrum

Esercizi pratici per entrare nel vivo della programmazione.

Lire 7.000

Clizio Merli

μPascal per Commodore 64/128

Un manuale completo per il programma compilatore

Lire 7.000

Umberto Colapicchioni e Luca Galuzzi

Dal registratore al drive del C64

Tutti i segreti delle memorie di massa del Commodore 64

Lire 7.000

Autori Vari

ADA

Il linguaggio passepartout dei computer degli anni '80.

Lire 5.000

Clizio Merli

Il linguaggio PASCAL

Un manuale tascabile per lo studio e la programmazione.

Lire 5.000

Sì, voglio arricchire la mia biblioteca con i seguenti volumi al prezzo di copertina + lire 3.000 per spese di spedizione.

- ☐ **64 Programmi per Commodore 64**
- ☐ **Strategie vincenti per i tuoi videogames**
- ☐ **62 Programmi per Vic 20 C16 e Plus77**
- ☐ **Utilities e giochi didattici**
- ☐ **Tutti i segreti dello Spectrum**
- ☐ **Simulazioni e test per la didattica**
- ☐ **Imparare giocando il basic dello Spectrum**
- ☐ **I miei amici C16 e Plus4**
- ☐ **Pascal per Commodore 128**
- ☐ **Dal registrattore al drive del C64**
- ☐ **ADA**
- ☐ **Il linguaggio Pascal**

Nome ....

via .... **N.ro.** .... **telefono** ....

CAP .... **Città** ....

Su tale importo mi praticherete lo sconto del 10% in quanto abbonato a ☐ *Commodore Computer Club* ☐ *Personal Computer* ☐ *Com puter* ☐ *VR Videoregistrare*. Pertanto vi invio la somma soltanto di lire ....

Valore dell'ordine lire....................

Ritagliare e spedire in busta chiusa regolarmente affrancata a Systems Editoriale - V.le Famagosta, 75 - 20142 Milano.

RAPPORTO SUI DIVERSI

Anche se non riuscite ad distaccarvi dal vostro amatissimo C/64, non potete fare a meno di considerare che, sull'altra sponda, esiste un lussureggiante mondo di sistemi informatici alternativi

di **Alessandro de Simone**

NON TUTTI SANNO CHE...

Ancora oggi è possibile osservare utenti sprovveduti che si disperano perchè il loro Olivetti M-24 non riesce a leggere dischetti contenenti programmi per C/64 (computer notoriamente meno "potente"); oppure non si spiegano coma mai il loro drive da 5 1/4 (cioè, "grosso") non solo non riesce a leggere dischetti da 3.5 (cioè, "piccoli") ma, addirittura, non è in grado di espellerli dopo averli introdotti nella fessura del drive a viva forza...

Scherzi a parte, siamo sicuri che i nostri lettori sanno benissimo la differenza tra un computer ed un altro, ma siamo altrettanto convinti che è bene ribadire alcuni concetti che, trascurati per troppo tempo, rischiano di farci allontanare dal mondo informatico, in costante ed inarrestabile avanzata.

In queste pagine ci intratterremo su alcuni concetti che, adeguatamente sviluppati, hanno consentito di arrivare al punto attuale della tecnologia che rappresenta null'altro che la base dei prossimi, irrinunciabili sviluppi dell'informatica.

ALL'INIZIO ERA IL L.M.

Qualsiasi unità di elaborazione, grossa o piccola che sia, home computer o mainframe o calcolatrice tascabile che trovate nei fustini di detersivo, possiede un unico, esclusivo ed insostituibile modo di operare: quello basato sul linguaggio macchina.

Qualunque sia la "struttura" informatica, sarà presente, sempre e comunque, un microprocessore, magari a 32 bit (o a soli 4 bit) che, in qualsiasi caso, mastica solo matematica binaria, basata su due soli simboli, lo zero (0) e l'uno (1).

Senza star qui a ricordare cose fin troppo note, ci limiteremo a suddividere i computer in varie categorie, ma solo dopo aver stabilito che esiste... una sola categoria di elaboratori(!).

Quando, insomma, accendiamo il terminale di una rete informatica (come quella, ad esempio della banca vicino casa), o il no-

Alla base di una qualsiasi procedura c'è sempre un gruppo di routine l.m.

LE AVVENTURE DI

PRIMO GIOVEDINI

by M. Mietta

"Tutta colpa del Virus"

Un programma utilizza spesso le risorse del computer: il S.O.

stro più modesto C/64, il microprocessore presente al suo interno viene "svegliato" ed inizia a pulsare, proprio come il cuore in una nuova creatura.

Un microprocessore possiede, al suo interno, un particolare circuito elettronico che gli consente di iniziare l'elaborazione di un programma che si suppone presente a partire da una certa locazione in poi. Senza programma di partenza (bootstrap) residente, è ovvio, su memoria Rom, nessun computer o nessun terminale potrebbe iniziare a funzionare, per quanto sofisticato sia.

Il circuito elettronico cui facciamo riferimento, e che DEVE esistere in ogni modello di microprocessore, fa capo al ben noto pin (piedino) di Reset; ciò significa che, in qualunque momento, è possibile "toccare" (elettronicamente) questo piedino per far ripartire l'elaborazione dall'inizio. Poichè all'accensione del computer non deve esser necessario toccare il tasto di Reset, un altro particolare circuito elettronico, basato sul caricamento di un condensatore, è realizzato in modo tale che, dopo un certo intervallo di tempo (idoneo ad attivare tutti i circuiti integrati, alcuni decimi di secondo) il pin di cui parlavamo venga "toccato" affinchè inizi l'elaborazione.

Dunque un qualsiasi terminale è dotato di un microprocessore, di almeno una Rom (la famosa Bios dei sistemi Ms-Dos) e di almeno una Ram, ma sappiamo che, ormai, vi sono vari mega di Ram, altro che almeno una!

A questo punto delle operazioni il computer è pronto per l'uso.

ROM E DISCHI

Le Rom di Bootstrap a volte sono costituite da poche migliaia di byte, altri computer, invece, sono gestiti da centinaia di migliaia di byte su Rom; quale è la differenza?

E' bene sottolineare, prima di individuare eventuali categorie, che un programma gira su un computer solo se sono disponibili tutte le risorse richieste; se, ad esempio, il nostro programma è in grado di disegnare a colori in hi-res o di inviare dati alla stampante, all'interno del computer, al momento dell'elaborazione, DEVONO esser presenti le corrispondenti routine l.m. in grado di gestire la pagina grafica (e la Ram ad essa relativa) e la procedura di comunicazione (comprensiva del protocollo di interfaccia stabilito). Se, poi, queste routine sono presenti su Rom oppure su Ram, non ha alcuna importanza per il microprocessore.

Se le routine sono presenti su Rom c'è un enorme vantaggio: in qualsiasi momento sono disponibili mediante semplici salti alle stesse Rom in modo diretto (le famigerate Sys del Basic) oppure "trasparente" cioè vengono attivate in modo indiretto mediante, ad esempio, un comando Basic. Alla categoria dei computer che dispongono del sistema operativo (S.O.) su Rom appartiene il C/64, il C/128 ed il Plus/4 che, addirittura, ha su Rom ben quattro programmi, pronti per l'uso.

Se, al contrario, le routine di gestione sono presenti su Ram, vuol dire che, quando il computer era spento, erano presenti altrove: sono state caricate in memoria, dopo aver acceso il calcolatore stesso, leggendole dal disco.

Questo appena visto è il caso di tutti i computer Ms-Dos compatibili (PC Commodore, Olivetti, IBM e tanti altri), dell'Amiga e di tutti i computer un po' più grossi di un semplice home.

Sembrerebbe che il secondo sistema (caricamento della maggior parte delle risorse dopo l'accensione del computer) sia più scomodo di quello residente su Rom, di-

Settembre 1988: sono già passate diverse settimane dal memorabile giorno in cui Primo Giovedini debellò i Galaxians, salvando il mondo intero da quella terribile minaccia e meritandosi così l'onorificenza "64K Ram"...

sponibile, invece, all'accensione della macchina.

Un S.O. su Rom permette di offrire elaborazioni molto semplici ed alla portata dei principianti; questo metodo è stato infatti molto apprezzato dalla utenza giovane che si è avvicinata per la prima volta ad un computer (Vic-20, C/64, C/128).

Se, però, nel corso degli anni ci si accorge che alcune risorse possono essere ottimizzate, l'unico modo per sostituire le vecchie routine con le nuove è quello di aprire il computer e sostituire la vecchia Rom con quella nuova, operazione non sempre gradita dagli utenti.

Ecco, quindi, il grande vantaggio offerto dal sistema operativo residente su Ram: in caso di aggiornamenti è sufficiente cambiar disco per disporre di nuove potenzialità. Questo è il sistema seguito da tutti i computer della fascia medio-alta, cioè quella dell'utenza cosiddetta "professionale".

LINGUAGGI

Allo stesso modo possiamo considerare i linguaggi residenti, o meno, su Rom. Il C/64, come è noto, contiene su Rom sia il S.O. che il Basic. I computer professionali, invece, nè l'uno nè l'altro.

Questo accade perchè non tutti i programmi commercializzati sono scritti in Basic e sarebbe sciocco tenere occupato uno spazio di memoria da un circuito integrato che rischia di rimanere inutilizzato. Quando, ad esempio, caricate un gioco in l.m. (di quelli che trovate in edicola) quasi nessuna routine della Rom del Basic viene utilizzata, mentre vengono attivate, quasi certamente, molte routine del S.O. (grafica, suono e così via). Il gioco, ovviamente, "gira" lo stesso ma, in ogni caso, in quel momento vi sono ben 8 KRom inutili, che potevano essere impiegati, ad esempio, per dotare il gioco di una schermata in più o di un effetto sonoro più sofisticato.

In un computer professionale, insomma, la disponibilità di grandi quantità di Ram consente un'efficienza ed una versatilità impossibili da superare con sistemi basati su Rom.

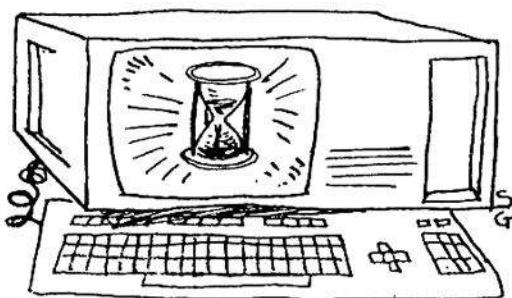
In ogni caso, e non ci stancheremo di ripeterlo, un programma (qualunque esso sia) gira grazie alla presenza di centinaia di routine scritte in linguaggio macchina, presenti su Rom oppure su Ram.

INTERPRETI E COMPILATORI

Stabilito, dunque, che un programma non è altro che un insieme di routine l.m. accuratamente interconnesse tra loro (e che, di tanto in tanto, "accedono" ad altre routine del S.O.), non rimane altro che approfondire la differenza esistente tra linguaggi interpreti e linguaggi compilatori che generano, rispettivamente, programmi interpretati e programmi compilati.

Un programma in Basic si dice interpretato perchè, per farlo funzionare, è necessaria la presenza contemporanea di una raccolta di routine (il linguaggio Basic, appunto) in grado di decodificare le varie istruzioni presenti nella varie linee di programma. Un programma scritto in Basic non può funzionare se prima non viene caricato il linguaggio Basic. Questo particolare spesso sfugge ai possessori di C/64 per il semplice motivo che il linguaggio è SEMPRE presente (in Rom) nel loro computer. Chi possiede un IBM compatibile, invece, si accorge più facilmente della differenza perchè se cerca di caricare solo un programma Basic, questo non gira.

Del resto anche i possessori di C/64, che hanno usato almeno una volta uno dei nu-

L'utente non può accorgersi se il programma che gira è scritto in Basic o in C

La velocità di elaborazione è uno degli indizi sull'origine del programma che vedete girare

merosi Tool disponibili (Simon's Basic, Toma ed altri) sanno che non è possibile far girare un programma contenente altre istruzioni se non si carica, prima, il linguaggio specifico o, quantomeno, le routine che consentono di gestire altri comandi Basic.

Un linguaggio compilatore, invece, consente di registrare un file (detto: oggetto) che può essere caricato e lanciato anche in assenza del linguaggio che lo ha generato. Un file compilato, infatti, è un programma in linguaggio macchina "completo" nel senso che contiene al suo interno tutte le routine che occorrono per il suo funzionamento, oppure fa ricorso alle sole routine del sistema operativo che si suppone, ovviamente, sempre presente nel computer.

Un programma scritto in Basic, insomma, non è altro che un insieme di codici da interpretare opportunamente da parte di una particolare procedura l.m. (contenuta all'interno dello stesso linguaggio interprete) che attiva, a sua volta, le varie routine l.m. in grado di produrre gli effetti desiderati.

Ne consegue che un programma Basic (o, comunque, uno interpretato) è più lento di uno compilato.

Vi sono in commercio, tuttavia, anche compilatori per il linguaggio Basic che, in pratica, "leggono" il programma, ne estraggono il significato più profondo e generano un programma autonomo, ricco di efficienti routine in linguaggio macchina, che risulta più veloce del programma Basic originario.

Il più noto dei compilatori per C/64 è l'Austrospeed, ormai di pubblico dominio, che consente di aumentare notevolmente la velocità dei programmi scritti in Basic. I file oggetto sembrano costituiti da una sola Sys che, in realtà, impartisce l'ordine di inizio della procedura l.m.

COMPILATORI E L.M.

La tendenza attuale del mercato spinge le software house a realizzare, da una parte, linguaggi compilatori, e dall'altra efficienti package in grado di compilare linguaggi interpreti.

La sfida procede di pari passo ed oggi è un po' difficile stabilire quale linguaggio (o procedura) sia più veloce, più semplice da usare oppure più "user friendly".

Nel campo Ms-Dos, ad esempio, notiamo la presenza di compilatori Basic molto più veloci, in certi casi, del sofisticatissimo Quick Basic, una sorta di interprete che, la momento della registrazione, compila il programma.

Sembra, tuttavia, che il futuro appartenga al "C" un linguaggio compilatore del resto piuttosto "duro" da digerire.

Un linguaggio equidistante dalla semplicità del Basic e dalla velocità del "C" è sicuramente, al giorno d'oggi, il Turbo Pascal, disponibile sia per sistemi Ms-Dos che su CP/M.

Entrambi i linguaggi, C e Turbo Pascal, nelle numerose versioni differenti tra loro sia per le macchine sulle quali girano sia per le potenzialità offerte, generano comunque file oggetto autonomi, in grado, cioè di esser caricati e lanciati in assenza dei linguaggi che li hanno generati.

Ciò è possibile, come già detto, grazie alla generazione automatica di routine in l.m. che sono in grado di soddisfare tutte le esigenze di programmazione desiderate.

SBAGLIANDO SI IMPARA

Continuano le considerazioni sugli errori più comuni che è possibile commettere in Basic

di **Alessandro de Simone**

Un errore si individua più facilmente se lo si commette a bella posta

Le variabili che è possibile "trattare" in Basic sono di tre tipi, intere, in virgola mobile e stringa.

La distinzione fra i tre tipi avviene grazie ad altrettanti simboli, o meglio, grazie a due caratteri speciali che li contraddistinguono.

Le variabili più usate, di solito, sono quelle in virgola mobile che possono essere rappresentate da due caratteri alfabetici oppure da un carattere alfabetico ed uno numerico. Esempi di variabili in virgola mobile sono i seguenti:

AA=12: VT=34: R5=-56.76

Le variabili intere hanno, come simbolo distintivo, il carattere di percentuale (%): dovrebbe esser noto che il loro valore può variare tra -32768 e 32767 (pena l'emissione di "Illegal quantity error"). Esempi:

AA%=12: VT%=34: R5%=-5676

Le variabili stringa, infine, sono contraddistinte dal simbolo del dollaro ($) posto dopo il loro nome. Esempi:

AA$= "Questo è un messaggio"
GH$= "Errore nei data"

Nel trattare le variabili numeriche, a volte è possibile commettere errori dimenticando di aggiungere (o di NON aggiungere) il simbolo di percentuale; ne consegue che, a volte, ci illudiamo di trattare un valore in virgola mobile mentre, in realtà, ci riferiamo ad un valore intero, o viceversa. Se, per esempio, vogliamo eseguire il prodotto tra la variabile in virgola mobile AA e quella intera BB% dobbiamo digitare...

AA=20: BB%=3: Print AA*BB%

...per avere il risultato desiderato (cioè 60).

```
100 REM ERRORE 22 TYPE MISMATCH
110 :
120 A$="QUESTA E' UN MESSAGGIO"
130 :    B$=LEFT$(A ,6)
140 REM B$=LEFT$(A$,6)
150 :     X =A$
160 REM:  X$=A$
170 :    READ A
180 REM READ A$:REM SYNTAX ERROR
190 DATA PAROLA:REM DATA N.1
200 READ A$
210 DATA 34     :REM DATA N.2
```

Se, però, dimentichiamo di apporre il simbolo di percentuale la prima volta, la seconda o entrambe le volte, come...

AA=20: BB=3: Print AA*BB%

```
100 REM ERRORE 23 STRING TOO LONG
110 :
120 A$=" DIGITA UN MESSAGGIO"
130 PRINT A$:INPUT A$:A= LEN(A$)
140 FOR I =1 TO 255:X=LEN(X$)
150 REM PROVA AD ELIMINARE RIGA 160
160 IF X+A > 255 THEN I=255:GOTO180
170 X$=X$+A$
180 PRINT X$: PRINT: PRINT: NEXT
190 PRINT "LUNGHEZZA:" LEN(X$)
```

Il comando Input riserva sorprese in più occasioni

...il risultato sarà zero, o comunque diverso da quello che ci attendevamo.

In questi casi, purtroppo, non viene segnalato alcun errore, ma l'elaborazione prosegue inesorabile e risulta difficile individuare l'incongruenza.

Nel caso, invece, in cui si faccia confusione tra variabile numerica (intera o virgola mobile) e stringa, il calcolatore se ne accorge e segnala prontamente un "Type mismatch error" (= errore di confronto di battitura).

```
100 REM ERRORE 23 (DISCO/NASTRO)
110 :
120 OPEN 1,8,8,"*,P,R"
130 INPUT#1,A$:PRINT A$;:GOTO130
```

Il programma n.1 genera, appunto, alcuni casi di errore dovuti a distrazione in fase di battitura. In riga 130, infatti, è presente la variabile A invece di A$ (come richiede la sintassi di Left$). Modificando la riga 130 come indicato in 140 si ovvia all'inconveniente.

A volte, però, un errore di "associazione" di un gruppo di caratteri alfanumerici ad una variabile numerica, anzichè ad una stringa, genera un Syntax error e non un Type mismatch come sarebbe logico aspettarsi.

Ce ne possiamo accorgere, dopo aver apportato le dovute correzioni alle righe 130 - 160, impartendo il Run ancora allo stesso programma; verrà visualizzato un Syntax error in 190, in corrispondenza, cioè, della riga Data che contiene il dato che risulta impossibile associare alla variabile numerica A.

Modificando la 170 come indicato in 180 si ovvia all'inconveniente. Un caso di dubbia soluzione, invece, rappresenta il valore 34 presente in 210 che può venir letto sia con un Read A che con un Read A$.

Per evitare dubbi sarà sempre bene inserire tra virgolette tutti i messaggi che si desidera far considerare come stringhe, in modo da far generare un Syntax error in caso di errore.

STRING TOO LONG

Dovrebbe esser noto che, nel Basic Commodore, è impossibile generare una stringa formata da un numero maggiore di 255 caratteri. Qualsiasi tentativo in contrasto con tale regola genera l'emissione del messaggio "String too long" che interrompe l'elaborazione in corso.

Il listato n.2 chiede di digitare una stringa che, in seguito, verrà sempre concatenata con se stessa aumentando, in tal modo, la sua lunghezza.

Se digitate il programma eliminando la riga 160 (oppre digitando un Rem prima di essa) noterete che, dopo un po' di tempo, l'elaborazizone viene interrotta con la visualizzazione del messaggio.

Il trucco per evitare tale inconveniente è, appunto, rappresentato dal "filtro" di riga 160 che, PRIMA di effettuare la concatena-

zione, effettua una somma tra le due stringhe in causa. Se la loro lunghezza è minore di 256 (cioè 255 o ancora minore) la concatenazione viene attuata, altrimenti viene evitata; naturalmente il lettore può intercettare tale situazione per far apparire un messaggio opportuno, oppure limitare la somma solo ad alcuni caratteri (mediante Mid$, Left$ e così via).

Una intercettazione impossibile, invece, è quella relativa al comando Input relativo al disco (o nastro) come si può studiare dal listato n.3.

In questo caso, infatti, l'Input di riga 130 è costretto ad accettare tutti i caratteri presenti sul file aperto, finchè non incontra l'opportuno delimitatore. Siccome, nella maggior parte dei casi (soprattutto con i file di programma) vi possono essere migliaia di caratteri diversi dal delimitatore richiesto, il numero massimo di 255 viene immediatamente raggiunto e superato, generando il messaggio.

Operando con il disco, quindi, è bene non usare mai il comando Input#, a meno che non si conosca esattamente la struttura del file; è sempre meglio, invece, riferirsi al comando Get# che, leggendo un carattere alla volta, è sicuramente più lento, ma evita di incorrere nell'errore citato (programma n.4).

CAN'T CONTINUE

Quando si interrompe un programma con il tasto Run/Stop è possibile riprenderne l'esecuzione con il comando Cont.

Se, però, dopo l'interruzione abbiamo generato un errore (digitando un tasto a casaccio e provocando un Syntax error) la continuazione viene impedita (Can't continue error : non posso continuare).

Sembrerebbe che questa limitazione sia una punizione troppo severa e che i progettisti avrebbero potuto essere più magnanimi. Tuttavia c'è da osservare che la generazione di un errore, occorsa dopo l'interruzione di un programma, potrebbe essere di lieve entità come di portata irrimediabile. Per sicurezza, quindi, è stato deciso di evitare la prosecuzione in qualsiasi caso.

```
100 REM ERRORE 23 (DISCO/NASTRO)
110 :
120 OPEN 1,8,8,"*"
130 GET#1,A$:IF ST<>0 THEN 200
140 IF A$= "" THEN 130
150 PRINT ASC(A$);: GOTO 130
200 CLOSE 1: END
```

```
100 REM ERRORE 26 CAN'T CONTINUE
110 :
120 A=25: B=3: C=45
130 PRINT "DIGITA: GOTO 150"
140 ERRORE INTENZIONALE
150 PRINT"I DATI CI SONO ANCORA!"
160 PRINT A*B/C: PRINTB/C
```

```
100 REM ERRORE 27 UNDEF'D FUNCTION
110 :
120 :    DEF FN AA(X)=SQR(X)
130 REM DEF FN  A(X)=X↑3
140 INPUT"RADICE QUADRATA DI";Y
150 PRINT "RADICE QUAD.:"  FN AA(Y)
160 PRINT "RAD.4TA" FN AA(FN AA(Y))
170 INPUT"CUBO DI";Y
180 PRINT FN A(Y)
```

Non utilizzate funzioni troppo complicate; spezzatele in più parti.

```
100 REM ERRORE 16 OUT OF MEMORY
120 :
130 A=1+(3-(3-(3-(3-(3-(3-(3-(3-(3-(3-(3)))))))))))
```

```
100 REM "SPEZZARE" UNA FUNZIONE COMPLESSA
110 :
120 REM INVECE DI...
130 :
140 A=LOG(SIN(.4*SQR(56-(LOG(63))/TAN(4↑(5)-SQR(45)*SIN(34-1.5)/4↑(4/7)))))
150 PRINT A
160 :
170 REM DIGITATE...
180 :
190 X=4↑(4/7)
200 Y=SQR(45)*SIN(34-1.5)
210 A=LOG(SIN(.4*SQR(56-(LOG(63))/TAN(4↑(5)-Y/X))))
220 PRINT A
```

Attenti a non confondere le variabili numeriche con le stringhe

Tuttavia è possibile aggirare l'ostacolo digitando, come prova, il listato n.5 che contiene un evidente errore di sintassi in riga 140.

La prima parte del listato assegna a tre variabili A, B, C altrattanti valori prima di interrompersi in riga 140. A questo punto è sufficiente digitare Goto 150 per rendersi conto che i valori sono ancora conservati in memoria e che l'elaborazione può esser portata a termine senza conseguenze. Indispensabile è ricordarsi di battere Goto 150 e NON Run 150, operazione, questa, che annullerebbe senza rimedio tutte le variabili fino a quel momento elaborate.

Se, dunque, incorrete in un qualsiasi errore, chiedete il listato "intorno" alla riga segnalata, prendete nota dell'errore (SENZA modificare la riga incriminata) e digitate il Goto opportuno, alla ricerca, magari, di altri errori.

Solo alla fine modificate le righe in cui avete notato gli errori prima di far ripartire il programma tenendo presente che non sempre il trucchetto suggerito può funzionare; soprattutto nei listati pieni zeppi di errori (di battitura e non).

UNDEF'D FUNCTION

Per ciò che riguarda lo studio di questo tipo di errore consigliamo ai lettori, per saperne di più, di leggere il manuale del computer in loro possesso.

Qui ci limiteremo a ricordare che è possibile definire una funzione alla quale far spesso riferimento nel corso di un programma.

Nella riga 120 viene definita, con il nome AA, la funzione, volutamente banale, che determina la radice quadrata di un numero. Si noti che, a prima vista, la variabile sia quella denominata con X mentre invece, come si può notare in riga 150 (Y), può avere nome diverso. Ciò significa che la variabi-

le che si desidera elaborare (Y, nel nostro caso) prenderà il posto di quella racchiusa tra parentesi (riga 120) e verrà elaborata come se quello fosse il suo vero nome; al termine, naturalmente, riprende il nome originario.

L'errore più comune che si può commettere operando con le funzioni è quello di richiamarle con un nome diverso da quello con cui sono state definite.

Il programma n.6, infatti, evidenzia tale "distrazione".

ANCORA FUNZIONI

Operando con funzioni complicate è possibile generare due tipi di errore. Il primo, paradossalmente, riguarda la memoria e lo possiamo notare digitando il programma n.7.

In questo caso l'elevato numero di parentesi genera un errore simile a quello relativo ad una nidificazione eccessiva di Gosub (già esaminato nel numero scorso) che visualizza un "Out of memory error" nonostante la Ram libera sia disponibile ancora in gran quantità.

In casi come quello pubblicato (volutamente semplice) è necessario "spezzare" la funzione su più righe Basic in modo da utilizzare, per ciascuna riga, un limitato numero di coppie di parentesi.

L'errore tipico di una funzione complessa, infatti, dovrebbe essere "Formula Too Complex error" che, tuttavia, è piuttosto difficile da realizzare appositamente dal momento che importanza notevole riveste l'intreccio di varie formule e funzioni tra di loro. Un paio di volte mi è capitato di osservare il messaggio citato (lavorando in Basic) ma ho dimenticato di prenderne nota...

Se, tuttavia, dovesse capitare a voi, ricordate che per ovviare all'inconveniente è sufficiente spezzare la formula in più parti. Ne è un valido esempio il listato n.8 la cui riga 140, a dispetto dalla sua complessità, non genera il messaggio in oggetto.

In ogni caso è fuori discussione che l'elaborazione di riga 140 si "legge" molto più facilmente se scissa nelle tre righe 190 - 200 - 210.

E se voi la leggete meglio, figuratevi un computer...

La massima lunghezza di una stringa è di 255 caratteri

UN DISCO TUTTO DA LEGGERE

Un nuovo comando Basic che consente di accedere facilmente ai testi archiviati su disco; ed una buona occasione per "scavare" più a fondo nel sistema operativo del C/128

di **Domenico Pavone**

Nei vostri programmi l.m. potete inserire "salti" al Kernal con la massima disinvoltura

Avendo la fortuna di disporre di un Basic 7.0 con ben 28K di ROM (pensate ai miseri 9K del C/64), può sembrare assurdo implementare nuovi comandi in una "bestiolina" come il C/128, ma è risaputo che non esiste essere più insaziabile del computeromane, e che... l'appetito vien mangiando.

Proverbi a parte, la routine proposta in queste pagine sfrutta un metodo abbastanza semplice per aggiungere dei comandi al Basic, fornendo inoltre un comodo strumento soprattutto per chi ha spesso a che fare con i word processor.

CHE COSA FA...

Per sfruttare l'utility, è sufficiente mandare in esecuzione (ovviamente dopo averlo copiato e salvato) il programma Basic pubblicato in queste pagine, che provvede a rendere operativa una routine scritta interamente in linguaggio macchina.

Da questo momento in poi è possibile digitare (direttamente o da programma) l'istruzione...

DPRINT "nome file"

...la quale leggerà, dal dischetto inserito nel drive, il contenuto del file indicato tra virgolette (a proposito, non dimenticate di chiuderle!), che dovrà necessariamente contenere un testo in codici ASCII, per mostrarlo sullo schermo.

Il nuovo comando può essere anche impartito nella forma abbreviata D? "nome file", e reagisce a tutte le facilitazioni dell'editor di schermo: tasto STOP per arrestare la visualizzazione, NO SCROLL per fermarne momentaneamente lo scorrimento, e Commodore per rallentarlo.

L'output è operante anche in 80 colonne (per chi dispone di un video adatto), e in tal caso si può attivare il modo FAST per rendere ancora più veloce la lettura, tuttavia già piuttosto spedita (siamo in l.m.!).

In caso di problemi con il DOS (file not found, drive not ready, e così via), la routine ne segnala il messaggio di errore, come pure se riscontra un errore nella digitazione del comando.

La maggiore comodità di DPRINT consiste nel poter leggere qualunque testo senza perdere ciò che si ha in memoria e, soprattutto, senza dover ricaricare il programma con il quale lo si è generato, word processor o data base che sia (anche un archivio dei vostri programmi può essere un file ASCII).

L'utility legge indifferentemente file sequenziali o file di programma, cosa che la

```
10 REM ----------------------------------------------------------------
12 REM          COMANDO <DPRINT> PER C=128 + DISK DRIVE
14 REM ----------------------------------------------------------------
16 :
18 SCNCLR:BANK15:FAST:FORX=0TO146:READA:B=B+A:POKE3072+X,A:NEXT
20 FORX=0TO7:READA:POKE996+X,A:B=B+A:NEXT
22 IFB<>18435THENSLOW:PRINT"ERRORE NELLE LINEE DATA!":END
24 SYS3072:SLOW:PRINT"COMANDO <DPRINT> ATTIVATO"
26 :
28 DATA 169,228,141,000,003,169,003,141,001,003,096,162,023,076
30 DATA 063,077,224,011,208,249,201,153,208,245,198,061,032,201
32 DATA 003,201,068,208,236,230,061,032,128,003,201,034,208,227
34 DATA 230,061,162,000,032,201,003,201,034,240,013,157,008,013
36 DATA 232,224,017,016,206,230,061,076,046,012,134,251,032,069
38 DATA 168,032,128,200,165,251,162,008,160,013,032,049,247,032
40 DATA 056,247,032,189,239,166,251,032,006,241,032,006,239,032
42 DATA 045,199,032,110,246,240,005,032,068,247,240,240,032,131
44 DATA 161,032,120,167,132,251,032,090,085,160,000,169,123,162
46 DATA 001,032,208,247,032,045,199,200,196,251,208,241,032,067
48 DATA 146,032,128,003,076,144,175,160,000,140,000,255,076,016
50 DATA 012
```

rende "indipendente" dal particolare software adottato per creare il file (non dimenticate, anche quello per C/64!), ma che presenta un'insidia: se infatti si legge un programma al posto di un testo, come risultato si avrà la comparsa di simboli incomprensibili, cambiamenti casuali dei colori dello schermo, e, anche se molto raramente, si può giungere all'inchiodamento della macchina.

In questo caso, la maggior parte delle volte sarà sufficiente premere Run / Stop + Restore per tornare alla normalità con ancora attivo il comando DPRINT; come ultima risorsa, non resterà che premere il tasto di reset e, se si vuole ancora adoperare D-PRINT, impartire nuovamente SYS 3072.

...E COME LO FA

Come si può notare dal disassemblato commentato, la routine è allocata in banco 15 a partire dalla locazione $C00 (decimale = 3072), area adibita a buffer di input per l'interfaccia RS-232, che quindi non dovrà essere usata per altri scopi.

Per implementare il nuovo comando, viene sfruttata la gestione degli errori da parte del C/128, che, di fronte ad una "sconosciuta" istruzione, come DPRINT, genererebbe inevitabilmente un Syntax Error.

Più in dettaglio, se durante l'esecuzione del testo Basic il sistema operativo incontra un errore, attiverà le seguenti procedure:
1) Deposita nel registro X il codice numerico corrispondente al tipo errore riscontrato

Con il C/128 risulta piuttosto agevole aggiungere nuovi comandi Basic

Nonostante il set di istruzioni Basic del C/128 sia molto esteso, potreste desiderare l'implementazione di altri comandi

(nel caso di Syntax error, $0B = decimale 11), mentre nell'accumulatore sarà presente l'ultimo carattere (o dato) del testo Basic esaminato.

2) Cede il controllo ad una sua routine posta all'indirizzo $4D3C (19772) di bank 15, che si concluderà con la visualizzazione sullo schermo dell'errore occorso.

In pratica, però, l'indirizzo $4D3C non è il vero e proprio ingresso alla routine di gestione degli errori: se infatti entriamo in ambiente Monitor, ed impartiamo il comando D F4D3C, vedremo che a questa locazione corrisponde JMP ($0300), cioè un salto all'indirizzo puntato dal vettore posto a $300 e $301.

L'indirizzo in questione è $4D3F, ma, come avrete capito, possiamo "dirottarlo" dove ci pare semplicemente modificando il contenuto del vettore, che è posto in RAM.

E' questo, appunto, il compito delle prime istruzioni della nostra routine (C00 - C0A), che fanno puntare il vettore a $3E4, area comune a tutte le configurazioni di banco, dove semplicemente si setta il bank 15 e si salta a $C10, ovvero alla routine vera e propria. Qui (seguite i commenti del disassemblato) viene esaminato il tipo di errore e, se non è un syntax e l'ultimo carattere Basic incontrato non è il token di Print, si fa continuare l'esecuzione alla routine di sistema posta a $4D3F (C10 - C16).

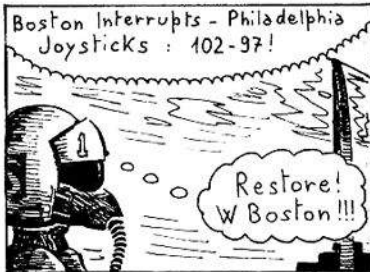
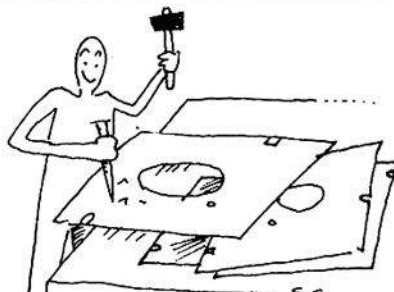
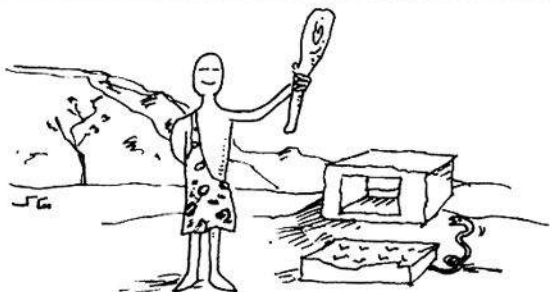
Le istruzioni successive controllano prima la presenza del carattere "D", poi delle virgolette ed infine, verificando che la sua lunghezza non sia superiore ai 16 caratteri (ecco perchè è necessario "chiudere" le virgolette), si legge il nome del file per copiarlo nelle locazioni da $D08 in poi (Errore 23 = String too long).

Queste operazioni vengono svolte sfruttando due routine di sistema che leggono il testo Basic, caricando nell'accumulatore il carattere esaminato: CHRGET ($0380) e INDTXT ($03C9). Entrambe usano come puntatore al testo Basic le locazioni $3D - $3E (61 - 62), ma mentre CHRGET "legge" il carattere successivo senza considerare la presenza di eventuali spazi, INDTXT riporta in accumulatore il carattere correntemente puntato da $3D - $3E. Importante da rilevare per non inficiare le successive operazioni, dopo l'esecuzione di queste routine il sistema rimane configurato in bank 14.

Da $C42 in poi viene massicciamente usato il kernal per leggere il contenuto del file e depositarlo sullo schermo (potete notare da soli gli indirizzi di salto descritti nel disassemblato), per concludere infine con un salto alla normale esecuzione del testo Basic ($C8D - $C90).

Unico punto forse degno di nota è la sezione della routine che si occupa di leggere gli eventuali errori del DOS ($C71 - $C8A), ovvero la variabile di sistema DS$. Il suo contenuto infatti è rintracciabile in bank 1, all'indirizzo puntato dalle locazioni di pagina zero $7B - $7C (122 - 124). Per mostrarlo sullo schermo, che risiede nei banchi caratterizzati da RAM 0, occorrerà quindi ricorrere a INDFETCH, la routine di sistema (con indirizzo $F7D0) che consente di leggere un carattere da qualsiasi banco, ponendo (prima del salto) in accumulatore il valore della locazione di pagina zero che punta al carattere da leggere (nel nostro caso, $7B), ed in X il numero di banco ove risiede (istruzioni $C79 - $C7F).

Il carattere letto si troverà in accumulatore, per cui risulterà poi semplice (istruzione $C82) visualizzarlo sullo schermo.

Con lo stesso sistema è possibile abilitare qualunque altro comando Basic, e se D-PRINT vi è apparso più complicato del previsto, nulla vieta di "farsi la mano" con più abbordabili cambi di colore dello schermo, vostri copyright o altro...

```
---------------------------DISASSEMBLATO D-PRINT PER C/128--------------

C00 LDA #$E4      Dirotta puntatori
C02 STA $0300     routine di
C05 LDA #$03      errore alla
C07 STA $0301     locazione $03E4.
C0A RTS
-------------------------------------
C0B LDX #$17      Errore 23.
C0D JMP $4D3F     Uscita a ERRORE.
-------------------------------------
C10 CPX #$0B      Sintax error?
C12 BNE $0C0D     Se no esci a ERRORE
C14 CMP #$99      Token di PRINT?
C16 BNE $0C0D     Se no esci a ERRORE
C18 DEC $3D       Punta a basic - 1.
C1A JSR $03C9     Legge testo corrente
C1D CMP #$44      E' uguale a "D"?
C1F BNE $0C0D     Se no esci a ERRORE
C21 INC $3D       Punta a basic + 1.
C23 JSR $0380     Legge testo seguente
C26 CMP #$22      Virgolette?
C28 BNE $0C0D     Se no esci a ERRORE
C2A INC $3D       Basic = basic + 1.
C2C LDX #$00      Azzera LEN file.
C2E JSR $03C9     Legge testo corrente
C31 CMP #$22      Virgolette?
C33 BEQ $0C42     Se si' vai a DPRINT
C35 STA $0D08,X   Copia nome file.
C38 INX           LEN file = LEN + 1.
C39 CPX #$11      Se LEN > 16, esci
C3B BPL $0C0B     con ERRORE 23.
C3D INC $3D       Punta a basic + 1.
C3F JMP $0C2E     Torna a leggere.
C42 STX $FB       LEN file in $FB
C44 JSR $A845     Setta bank 15.
C47 JSR $C880     Passa a minuscolo.
C4A LDA $FB       Prepara
C4C LDX #$08      parametri
C4E LDY #$0D      del file e...
C50 JSR $F731     Set Name.
C53 JSR $F738     Set File.
C56 JSR $EFBD     Open File.
C59 LDX $FB       Nr. logico file.
C5B JSR $F106     File in Input.
C5E JSR $EF06     Leggi Carattere.
C61 JSR $C72D     Stampa Carattere.
C64 JSR $F66E     STOP Premuto?
C67 BEQ $0C6E     Se si' vai a FINE.
C69 JSR $F744     Leggi STATUS WORD.
C6C BEQ $0C5E     Se ST=0, ripeti.
C6E JSR $A183     Chiudi File.
-------------------------------------
C71 JSR $A778     Leggi DS$.
C74 STY $FB       LEN(DS$) in $FB.
C76 JSR $555A     PRINT.
C79 LDY #$00      Offset.
C7B LDA #$7B      Punta a DS$.
C7D LDX #$01      Bank 1.
C7F JSR $F7D0     IND FETCH.
C82 JSR $C72D     Stampa Carattere.
C85 INY           Ripeti per tutti
C86 CPY $FB       i caratteri
C88 BNE $0C7B     di DS$.
C8A JSR $9243     Resetta DS$.
-------------------------------------
C8D JSR $0380     Punta a basic + 1.
C90 JMP $AF90     Esegui il basic.

------------------------------------------------------------------------

3E4 LDY #$00      Configura sistema
3E6 STY $FF00     in bank 15.
3E9 JMP $0C10     Salta a routine.
```

LO STRIZZACERVELLI

Come costringere il vostro C/64 ad interpretare, in modo totalmente automatico, qualsiasi nuovo comando di vostra creazione

di **Giancarlo Mariani**

Il C/64 interpreta un comando Basic secondo una procedura ben definita

Il C/64, come lo era stato a suo tempo il Vic 20, è un computer che, per la sua fascia di prezzo, offre ottime prestazioni che permettono lo sviluppo di giochi, o programmi di utilità, con effetti grafici e sonori piuttosto efficaci.

La limitazione principale di questi computer risiede nel Basic 2.0 che, con le sue poche istruzioni, deve far di tutto per gestire grafica, suono, sprite, e qualsiasi altra "cosa" che non rientri nel Basic standard.

Per gestire le interessanti caratteristiche del C/64, quindi, bisogna ricorrere ad una marea di Poke e Peek, che, anche se offrono il risultato desiderato, creano confusione nel listato Basic. Sarebbe bello, invece, avere istruzioni dedicate per cambiare, ad esempio, colore allo schermo, invece delle noiose Poke 53280 e 53281; oppure per muovere gli sprite, o per disegnare in grafica, e così via.

A queste cose hanno già provveduto altri, prima di noi, non ultima la Commodore stessa, che ha proposto, tempo addietro, programmi come il Simon's Basic, o dotando il C/128 del potente Basic 7.0. Per non parlare del "nostro" Danilo Toma che, con le sue routine grafiche, ha permesso a molti utenti di conoscere le tecniche per realizzare disegni "elettronici".

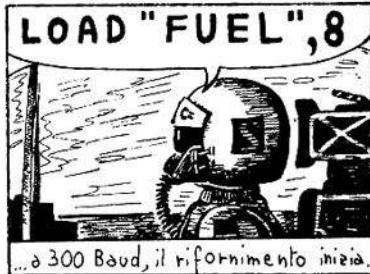
Tutti questi programmi, e moltissimi altri, sono interessanti ed utili, ma, proprio perchè sono "blocchi unici", non consentono una semplice personalizzazione di nuovi comandi.

In queste pagine illustreremo il modo per aggiungere un qualsiasi comando Basic attraverso un programma misto Basic-l.m. che permetta di aggiungere, con una certa facilità, tutti i comandi Basic desiderati, anche A CHI DI LINGUAGGIO MACCHINA NON CAPISCE ASSOLUTAMENTE NIENTE.

Ma cominciamo dall'inizio, spiegando come è possibile far riconoscere al C/64 un nuovo comando Basic e, quindi, eseguirlo.

LA TEORIA PRIMORDIALE

Il Sistema Operativo (S.O.) del C/64 contiene svariate routine su Rom che permettono lo svolgimento del compito appena descritto. I passi che compie per eseguire un'istruzione sono i seguenti:

1- Al momento della digitazione di una riga Basic, il S.O. tokenizza (ossia converte in un codice di un solo byte) le varie istruzioni Basic che compongono la stessa linea, e le memorizza nella Ram destinata ad ospitare la riga. Questo sistema consente di avere in memoria qualsiasi comando Basic sotto forma di un codice di un solo byte; in conseguenza di ciò si ottiene un notevole risparmio di memoria perchè, ad esempio, se non venisse convertito, il comando RESTORE,

invece di un solo byte ne occuperebbe sette, tanti quanti sono, cioè i caratteri che lo compongono.
2- Al momento dell'escuzione del programma (quando si digita Run) l'interprete "legge" il programma Basic, ne "estrae" i singoli byte-codice ed esegue l'istruzione, corrispondente ai vari codici trovati, attivando l'opportuna routine l.m.

Per aggiungere istruzioni nuove si potrebbe quindi usare il sistema della tokenizzazione, utilizzando codici non occupati nel C/64 (ed infatti così è stato fatto nel nostro emulatore GW-Basic). In questo modo, però, bisognerebbe riscrivere la routine di tokenizzazione, e questa non è una cosa molto semplice nè automatizzabile.

Siccome, al contrario, vogliamo rendere le vita più semplice possibile, è meglio non far tokenizzare i nuovi comandi, ma farli "riconoscere", carattere per carattere, al momento dell'esecuzione.

Spieghiamoci meglio: la routine che esegue il compito spiegato prima al punto 2, legge dapprima il programma Basic; se trova un codice (token) che conosce, passa all'esecuzione del comando specifico. Se quindi noi, invece di far saltare il C/64 a quella routine, lo facessimo "passare" dapprima attraverso una nostra routine, che legga e riconosca i nuovi comandi, il gioco sarebbe fatto.

Questo è più semplice di quel che si pensi, dal momento che il C/64, per saltare alla routine incriminata (che chiameremo d'ora in poi "Execute statement", Esegui comando), usa due locazioni Ram, che contengono, rispettivamente, il byte basso ed il byte alto dell'indirizzo di partenza della routine citata.

Queste due locazioni sono $0308 e $0309 (776 / 777). Il valore standard per $0308 è $E4, mentre per $0309 è $A7; ne consegue che la Execute Statement si trova posizionata, nelle Rom, a partire da $A7E4.

Sembrerebbe che sia sufficiente alterare le suddette locazioni in modo da farle puntare alla nostra routine di interpretazione. Per fare ciò, però, NON si possono impartire semplicisticamente due POKE "dirette"; infatti il Basic eseguirebbe dapprima la prima Poke, che verrebbe eseguita correttamente; per interpretare la seconda Poke, invece, dovrebbe, di nuovo, interpretarla attraverso la routine il cui indirizzo è ancora contenuto in 776 / 777. La prima delle due locazioni, però, è stata appena modificata dalla prima Poke, ed il sistema si inchioderebbe a causa di un salto ad una routine sconosciuta.

L'unico modo per cambiare le due locazioni, senza correre rischi, è quello di eseguire una micro routine in l.m. che, tramite due LDA e due STA, posiziona i puntatori come desiderato.

Chiamando con X l'indirizzo di partenza della nostra nuova routine di Execute Statement, in $0309 andrà inserito il valore...

INT (X/256)

...mentre in $0308 il valore:

X-INT(X/256)*256

Poichè il computer, oltre a riconoscere i nostri nuovi comandi, deve ovviamente eseguire anche tutti quelli "vecchi", bisognerà fare in modo che, passando attraverso la nostra routine, vengano riconosciuti anche i comandi standard del C/64. Il sistema più semplice per ottenere questo scopo è il seguente:

Durante lo svolgimento di un qualsiasi comando (posto in un programma Basic o impartito in modo diretto), il C/64 usa due locazioni di pagina zero per "ricordare" a che

E' sufficiente "imitare" la Rom del Basic per disporre di nuovi, potenti comandi

I nomi dei nuovi comandi devono esser sempre preceduti dal carattere di freccia a sinistra

punto si trovava al momento in cui ha interpretato il comando.

Queste due locazioni, che sono la $7A e la $7B, contengono proprio l'indirizzo da cui prelevare il carattere che il C/64 deve leggere in quel momento. Ad esempio, se impartiamo in modo diretto un'istruzione del tipo...

Print A,"Ciao"

...all'inizio $7A e 7B saranno posizionati sul carattere "P", poi su "R", poi su "I", e così via, fino all'ultimo carattere del comando, compresi i parametri. Alla gestione dei puntatori provvede automaticamente una routine l.m. posta a partire dall'indirizzo $0073, che ha il compito di incrementare i puntatori e leggere il carattere corrispondente alla posizione di questi ultimi.

Ovviamente la Rom contiene, al suo interno, alcune tabelle che consentono il riconoscimento delle varie istruzioni. La tabella delle parole Basic è posta a partire dall'indirizzo $A09E. Andando a curiosare con un monitor di l.m. si scopre che in questa tabella sono contenute, una dopo l'altra, tutte le parole chiave del Basic del C/64, carattere per carattere, e che il carattere finale di ogni comando è SHIFTato (cioè maiuscolo), per permettere, al S.O. di riconoscere la fine di ogni comando. Ad esempio il comando DATA sarà memorizzato nella tabella come "datA". In fondo alla tabella è posto uno zero ($00), che informa il C/64 della fine delle parole chiave del Basic.

Associata alla tabella delle parole, esiste anche un'altra tabella, denominata "Tabella degli indirizzi", posta da $A00C, che contiene gli indirizzi di partenza (decrementati di uno) di tutte le parole chiave del Basic. Gli indirizzi sono inseriti nella tabella sotto forma di Byte basso - Byte alto, e sono nello stesso ordine dei comandi della tabella delle parole.

Al primo comando della tabella delle parole, ad esempio, corrisponderà l'indirizzo contenuto nei byte 1 e 2 della tabella degli indirizzi, al secondo comando i byte 3 e 4, e così via.

In definitiva, il riconoscimento di un comando in modo diretto (al momento della pressione del tasto Return) avviene nel modo seguente:

1- Il S.O, tramite la routine $0073, legge un byte dalla riga di schermo (che contiene, presumibilmente, comandi ed istruzioni Basic) e lo confronta con il primo carattere della tabella delle parole.

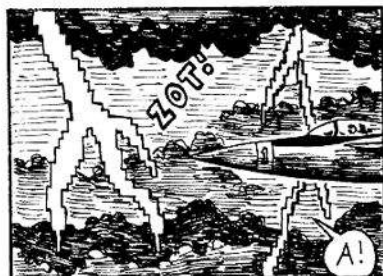
2- Se questi caratteri non sono uguali, rilegge lo stesso carattere del punto 1, e lo confronta con il primo carattere del successivo comando della tabella (il carattere, cioè, posizionato dopo l'ultimo Shiftato)

3- Se questi non sono uguali, ritorna al punto 2.

4- Se i caratteri sono uguali, continua a leggere e a confrontare, carattere per carattere il comando contenuto nel rigo di schermo con quello della tabella delle parole.

5- Se si arriva all'ultimo carattere (quello shiftato) e tutte le lettere sono uguali, legge l'indirizzo di partenza del comando dalla tabella degli indirizzi, e quindi salta all'esecuzione del comando.

6- Se, invece, anche uno solo dei caratteri incontrati non corrisponde a quello riportato in tabella, salta al punto 2.

7- Se, durante tale fase di controllo, viene individuato il byte $00 nella tabella delle parole, significa che è stata raggiunta (come detto in precedenza) la fine della tabella senza aver trovato alcun comando valido; quindi controlla se il comando impartito è il nome di una variabile da assegnare, e in caso negativo, visualizza il messaggio "Syntax Error".

UNA NUOVA TABELLA

Per riconoscere nuovi comandi, quindi, avremo bisogno di una tabella delle parole, di una tabella degli indirizzi aggiuntiva, e di una apposita routine che provveda a leggerle.

La tabella da costruire per usufruire della routine di queste pagine, deve essere così strutturata:

- le parole chiave vanno inserite una dopo l'altra, con l'ultimo carattere shiftato
- tra una parola e l'altra va inserito l'indirizzo di partenza (in forma byte basso - byte alto) della parola appena scritta.
- Alla fine della tabella va posto uno $00.

Supponiamo, ad esempio, di voler aggiungere due nuovi comandi Basic, Color e Plot, le cui corrispondenti routine l.m. iniziano da $2530 (Color), e da $8760 (Plot): in questo caso la tabella sarà così composta:

coloR3025ploT608700

Come si vede, infatti, dopo ogni comando (che DEVE avere l'ultimo carattere shiftato, ossia il suo codice deve essere uguale a quello del carattere "normale" più $80) è scritto il corrispondente indirizzo di partenza (ricordate, nella forma byte basso- byte alto, cioè "capovolto"); alla fine della tabella, lo ribadiamo, deve esser posto uno zero.

Per semplificare la routine di lettura, l'intera tabella può essere lunga al massimo 255 caratteri (se cercate di inserire tabelle più lunghe il sistema va in tilt); la tecnica descritta in queste pagine, quindi, utilizza una sola tabella invece di due (come fa, invece, il S.O.). Questo particolare riduce, di fatto, il numero dei nuovi comandi che possono essere aggiunti, ma ha il vantaggio di semplificare la routine di lettura e di esecuzione dei nuovi comandi.

La procedura, insomma, funziona in un modo assai simile a quella del S.O; in altre parole, modificando i puntatori ($0308 / $0309), la nuova routine di riconoscimento verrà eseguita ogni volta che il C/64 incontrerà una qualsiasi sequenza di caratteri, PRIMA della "vera" Execute Statement, permettendo così l'interpretazione delle nuove parole chiave.

La procedura per implementare i nuovi comandi è resa totalmente automatica dal programma di queste pagine

COME FUNZIONA

1- Come prima cosa, la routine memorizza i puntatori del Basic ($7A / $7B) nelle locazioni $FA / $FB, in modo da averli sempre disponibili. Ricordiamo che, all'atto della chiamata della routine puntata da $0308 / $0309, i suddetti puntatori sono sempre posizionati sul primo carattere di un comando posto in un programma Basic o impartito in modo diretto.

2- In seguito, tramite la routine $0073 (Read next character), la routine di queste pagine legge il carattere dal comando Basic e lo confronta con la nuova tabella delle parole.

3- Se viene trovato un carattere diverso, vengono ripristinati i puntatori da $FA / $FB, e ricomincia la lettura (punto 2) dal successivo comando della nuova tabella.

4- Se nella nuova tabella viene letto uno zero ($00) come primo carattere di un comando, significa che si è giunti alla fine; pertanto la procedura rimette a posto $7A / $7B (da $FA / $FB) e salta alla normale Execute Statement del Basic (posta a $A7E4), in modo che il S.O. possa riconoscere anche i comandi standard.

5- Se il confronto di tutti i caratteri fornisce esito positivo, allora si passa alla lettura degli indirizzi (posti nella nuova tabella posta

Ma un pericolo ancora maggiore minaccia Primo...

A! Ed ora cosa sta succedendo?

Una qualsiasi routine l.m. può essere trasformata in comando Basic, purchè termini con l'istruzione RTS

subito dopo il comando) ed all'esecuzione del comando stesso.

Per semplificare, e velocizzare, la gestione dei comandi, nella routine vengono utilizzati due accorgimenti:

Il primo è che ogni comando nuovo, sia inserito in un programma Basic che impartito in modo diretto, deve essere preceduto dal carattere di freccia a sinistra (il primo tasto in alto a sinistra della tastiera). Questo "trucchetto" permette di velocizzare il riconoscimento dei nuovi comandi, perchè se il primo carattere letto è, appunto, una freccia a sinistra viene attivata la nuova routine; in caso negativo rimette subito a posto i puntatori $7A / $7B e passa direttamente alla normale execute statement del S.O. ($A7E4), evitando, pertanto, il confronto delle parole carattere per carattere e risparmiando un notevole lasso di tempo.

Il carattere di freccia a sinistra, sebbene vada scritto come primo carattere di ogni nuovo comando, NON va messo nella nuova tabella delle parole, pena il non riconoscimento del comando stesso. Nella tabella andranno quindi trascritti i comandi così come sono.

Il secondo accorgimento riguarda un problemino nelle routine l.m. atte a servire i nuovi comandi implementati. Le routine, infatti, per funzionare correttamente, non devono finire (ossia tornare al Basic) con un RTS, ma con un salto (JMP) a $A7AE. Tale routine, denominata "Execute next statement", si occupa, appunto, di leggere ed eseguire il successivo comando incontrato in modo diretto o in modo programma. Se non venisse attivato il salto alla fine della routine, quest'ultima funzionerebbe correttamente, ma al momento del ritorno al Basic possono succedere cose spiacevoli, quali strani Syntax Error, oppure il non riconoscimento dei comandi successivi separati dal carattere di doppio punto (:), o cose del genere.

Per evitare all'utente il noioso compito di inserire il salto alla fine di ogni sua nuova routine, si ricorre ad uno speciale accorgimento: alla fine della routine (come si può notare nel disassemblato), tramite due PHA, si carica nello stack pointer l'indirizzo della routine $A7AE (decrementato di uno) prima di effettuare il salto alla routine che serve il nuovo comando; in questo modo, al ritorno al Basic da quest'ultima, il C/64 "prende" gli indirizzi di ritorno posti in precedenza, e quindi salta alla $A7AE.

La tecnica descritta consente di utilizzare una qualsiasi routine l.m, come comando nuovo, senza apportare alcuna modifica: anzi, queste routine DEVONO terminare con un RTS per assicurarne il corretto funzionamento.

Ricapitolando, i passi da fare per "costruire" nuovi comandi Basic sono i seguenti:

1- Trascrivere (o leggere) in una qualsiasi parte di memoria (dato che è RILOCABILE) la sola nuova routine di Execute Statement, tralasciando la parte che sposta i puntatori $0308 / $0309 e quella che esegue il Save della memoria.

2- Costruire (o leggere) la nuova tabella delle parole secondo il formato descritto prima: nome comando1 con l'ultimo carattere shiftato, relativi indirizzi LO/HI, nome comando2 con l'ultimo carattere shiftato... e così via fino alla fine, non dimenticando di inserire il carattere nullo ($00) in coda.

3- Modificare, nella nuova execute statement, l'indirizzo di lettura della nuova tabella. Questo si può trovare nel disassemblato ed è usato in vari punti della routine: ricordatevi di MODIFICARLI TUTTI.

4- Scrivere (o leggere) nella zona di memoria opportuna tutte le routine corrispondenti ai nuovi comandi aggiunti.

5- Modificare (tramite una microroutine l.m. del tipo di quella nel disassemblato) i puntatori $0308 / $0309 posti all'inizio della nuova execute statement.

A questo punto siete pronti per usare i nuovi comandi e, ammettendo di non aver commesso errori, il tutto dovrebbe funzionare correttamente. Ovviamente dovrete ricordare di non usare comandi che si sovrappongano in qualche modo alla routine o alla tabella, o che utilizzino le locazioni in cui sono posizionate queste ultime; infine ricordate di proteggere la zona di memoria, in cui sono allocati i nuovi comandi, tramite la modifica dei puntatori 55 / 56.

IL PROGRAMMA BASIC EXPANDER

Il procedimento descritto sinora, sebbene sia relativamente semplice, non è facilmente comprensibile a chi di l.m. capisce poco o niente; proprio per questo scopo (ma anche per gli "smanettoni" che non hanno voglia o tempo di far "fatica"), è stato realizzato un programma misto basic-lm, denominato "Basic Expander", che consente di fare tutte le operazioni descritte in modo completamente automatico, ossia di aggiungere nuovi comandi semplicemente specificandone il nome e l'indirizzo di partenza.

Il programma, in seguito, provvede a codificare la tabella, a segnalare eventuali errori commessi nell'immissione della stessa, a "leggere" la nuova routine di execute statement, a posizionare correttamente la lettura della nuova tabella dei comandi, ed infine a salvare il tutto su disco, in modo che possa essere richiamato tramite un paio di Load e azionato tramite una sola SYS.

Chi lavora con il solo registratore (ma che aspettate a comprare un drive?) dovrà faticare di più in termini di tempo e pazienza.

Ma vediamo come utilizzare la procedura:

Dopo aver digitato il programma di queste pagine, consigliamo di controllarlo più di una volta perchè, specialmente nei dati l.m. e nelle subroutine di codifica tabella e salva l.m, bisogna assolutamente evitare di commettere errori, pena l'inchiodamento totale oppure un più subdolo malfunzionamento semi-casuale.

E' consigliabile salvare sempre il programma su memoria di massa PRIMA di dare il Run, per evitare di doverlo ridigitare nel caso succedesse qualche imprevisto. Se, poi, non dovesse funzionare come vi aspettate, consigliamo CALDAMENTE di spegnere il computer, RIACCENDERLO e caricare la versione precedentemente registrata: non spegnendo il computer, infatti, alcuni puntatori potrebbero restare alterati in modo scorretto con CONSEGUENTI MALFUNZIONAMENTI DIFFICILISSIMI DA RINTRACCIARE IN SEGUITO.

Solo rispettando i consigli precedenti potrete evitare di impazzire come un cavallo. Dopo il Run compare una schermata con le opzioni previste ed i vari indirizzi delle routine che rappresentano gli indirizzi di partenza e di fine della nuova execute statement e della nuova tabella delle parole. Poichè il programma, automaticamente, riloca sia l'una che l'altra a seconda degli indirizzi di partenza, questi possono essere cambiati a piacimento tramite l'opzione 4 (modifica indirizzi).

E' da notare che tramite questa opzione possono essere cambiati solo i due indirizzi di partenza, dal momento che quelli di fine vengono posti automaticamente alla partenza + 123 byte per quanto riguarda la routine l.m. e alla partenza + 255 byte per la tabella.

Ricordatevi di alterare il Top di memoria, pena il probabile inchiodamento del sistema

Tutte le routine dell' enciclopedia possono esser trasformate in comandi Basic

La prima operazione da fare (dopo aver eventualmente modificato gli indirizzi) la prima volta che si fa girare il programma dovrebbe essere la numero 3, ossia "Modifica tabella". Tramite questa opzione è possibile aggiungere, cancellare, modificare qualsiasi comando che voglia essere inserito nella tabella. Scegliendo l'opzione 3 e premendo Return, il C/64 pone la domanda "Dal Numero:". Andrà digitato il numero del comando da inserire o modificare nella tabella. Tale numero non ha effetti pratici nella codifica dei nuovi comandi, ma, dato che la tabella è gestita tramite array, esso corrisponde all'indice dell'elemento dell'array, ossia, in pratica, al numero progressivo associato ad ogni comando.

Se la vostra mente comincia a vacillare, fermatevi un attimo, prendete un bel caffè (il decimo) e rileggete questo paragrafo.

Questo numero, dicevamo, la prima volta che si fa girare il programma deve essere 1.

Dopo aver premuto Return, il programma chiede "Al numero:". Inserite il numero dell'ultimo comando da modificare.

Ad esempio, se nella tabella abbiamo 15 comandi, dovremo inserire "...dal numero 1 al numero 15". Ovviamente nessuno vieta di inserire limiti minori o maggiori di questi, per modificare parzialmente la tabella o aggiungere nuovi comandi alla stessa.

Se in memoria non abbiamo alcuna tabella, questi limiti andranno inseriti a seconda del numero dei comandi da inserire nella tabella stessa. Ad esempio, se prevediamo di inserire 10 nuovi comandi, dovremo immettere "...dal numero 1 al numero 10".

Per problemi di visualizzazione, questi due numeri non possono differire di più di 20: se, quindi, nella tabella abbiamo 35 comandi, dovremo prima inserirli dal numero 1 al numero 20, e poi richiamare di nuovo l'opzione inserendo gli altri, dal numero 21 al numero 35.

Una volta inseriti i limiti, il C/64 mostrerà una schermata nella quale verranno visualizzati tutti i comandi (e relativi indirizzi di partenza) presenti nella tabella compresi tra i due limiti indicati. Nel caso in cui la tabella sia vuota (o che i limiti eccedano il numero di comandi presente) verranno visualizzati solo spazi vuoti al posto del comando e del corrispondente indirizzo.

A questo punto, nella penultima riga di schermo, viene visualizzata una riga, che copia la prima riga della tabella visualizzata, e viene posizionato il cursore all'inizio della riga stessa. Questo accorgimento permette la modifica o l'inserimento di un comando nella tabella; alla pressione del tasto Return si passerà alla modifica / inserimento del relativo indirizzo, visualizzato accanto al nome del comando stesso. Alla successiva pressione del tasto Return, verrà visualizzato il successivo comando da inserire (o modificare), il cursore verrà posizionato all'inizio dello stesso, e così via, fino al termine dei comandi visualizzati, istante in cui il programma torna al menu principale.

Se, come primo carattere del nome del comando, viene inserito l'asterisco (*), la fase di inserimento si arresta ed il programma torna al menu principale. Se, invece, viene inserito il segno uguale (=), il comando visualizzato viene cancellato, ed i comandi posti dopo quello cancellato vengono "spostati indietro" di una posizione nella tabella.

Quando la fase di inserimento della tabella è finita, si potrà controllare la correttezza della stessa, di nuovo tramite l'opzione 3 che, come detto prima, visualizza subito tutti i comandi entro i due limiti specificati. Nel caso si rilevino errori nella tabella sarà sufficiente premere il tasto Return un numero di volte adeguato in modo che il C/64 visualiz-

zi il comando da modificare, quindi apportare le correzioni e premere Return.

Se invece tutto risulta in ordine (ed in ogni caso, quando si conclude la fase delle correzioni) basterà scrivere un asterisco e premere return per ritrovarsi nel menu principale.

Dopo aver controllato che la tabella non contiene errori, la si potrà salvare su disco tramite l'opzione 2 (Salva tabella). Questa salva la tabella su disco sotto forma di file sequenziale (che quindi può essere letto anche tramite un Word Processor del tipo "Easy Script") così come è stata scritta dall'utente.

Anche questa opzione prevede l'inserimento dei due limiti, in modo da consentirne il salvataggio parziale.

La tabella può essere riletta per usi futuri o a successive ri-partenze del programma tramite l'opzione 1 (leggi tabella). Tale opzione carica da disco la tabella scritta tramite l'opzione 2, leggendo automaticamente il giusto numero di comandi salvati.

Sia l'opzione 1, che la 2, chiedono il nome del file. A questo nome verrà automaticamente aggiunta l'estensione ".TAB", per differenziare il file da altri programmi o file presenti sullo stesso dischetto. Alla fine delle operazioni di salvataggio, o caricamento, della tabella verrà stampato lo status del disco, in modo da controllare che, nei due casi, sia andato tutto bene.

L'opzione 5 (Salva l.m.), è la più importante del programma perchè è quella che si occupa della codifica della tabella, della lettura della routine l.m, del controllo di eventuali errori presenti, ed infine del salvataggio della tabella e della routine come programmi l.m.

Una volta richiamata l'opzione, il programma chiederà i "soliti" due limiti, e quindi passerà alla fase di formazione della tabella, prendendo i comandi compresi tra i due limiti specificati.

Se il nome di un comando comprende caratteri non alfabetici (non compresi tra "a" e "z") oppure il relativo indirizzo è minore di zero o maggiore di 65535, verrà visualizzato il messaggio "Comando non inserito:" con il relativo comando ed indirizzo errati.

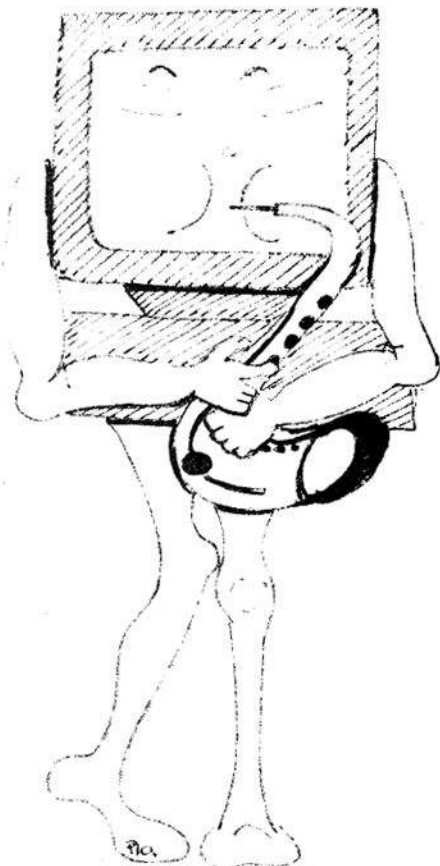
Questo errore non è distruttivo, ossia il programma continua ugualmente nella creazione della tabella limitandosi ad ignorare i comandi incriminati. Sarà poi discrezione dell'utente decidere se salvare ugualmente, o meno, la tabella priva dei comandi erroneamente digitati.

Se, invece, nella creazione della tabella il programma si accorge che questa supera i 255 byte di lunghezza, la fase di creazione si arresta, viene segnalato l'errore e viene chiesto di premere Return per tornare al menu iniziale. Dopo un errore del genere, ovviamente, tramite l'opzione 3 si dovranno cancellare alcuni comandi in modo da far rientrare la lunghezza della tabella nel limite consentito.

Dopo un certo tempo (che dipende dal numero dei comandi inseriti) in cui il C/64 completa la tabella, viene visualizzata la domanda "Nome:". Qui va digitato il nome del file con cui vogliamo salvare le routine dei nuovi comandi.

A questo punto il programma salva DUE file distinti:

- Il primo, con il nome formato da quello inserito più il suffisso ".R", l'indirizzo di partenza della routine (ad esempio, "COMD1.R832), che contiene la routine l.m. di interpretazione dei nuovi comandi.
- Il secondo, con nome formato da quello inserito più il suffisso ".T" e l'indirizzo di partenza della nuova tabella delle parole (ad esempio, "COMD1.T52992") che contiene

Potete creare un'infinità di "dialetti" Basic: l'unica limitazione è la vostra fantasia!

la tabella dei nuovi comandi Basic.

Dopo ogni salvataggio viene visualizzato lo status del disco, in modo da controllare che tutto sia andato bene.

Su disco, insomma, si troveranno due file distinti, che contengono tutto il l.m. che occorre per azionare i nuovi comandi.

Per usufruire di questi ultimi bisognerà, innanzitutto, caricare in memoria le routine che formano i nuovi comandi, quindi caricare il file ".R",8,1 (nell'esempio di prima, Load "COMD1.R832",8,1) ed il file ".T",8,1 (Load "COMD1.T52992",8,1). Ricordate, a questo punto, di "abbassare" il top di memoria, se necessario, tramite le locazioni 55 / 56, e di impartire una SYS all'indirizzo di partenza della nuova routine di execute statement (nell'esempio, SYS832).

Ora siete pronti per usare i nuovi comandi. Ricordatevi di scrivere la freccetta a sinistra come primo carattere di ogni comando e, nel caso mettiate un nuovo comando dopo il THEN, di separarlo con il carattere di doppio punto (:). Esempio:

If a=1 then :<-- color 2

...e non...

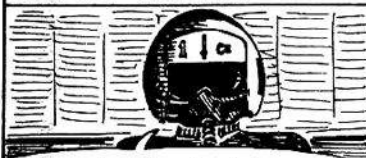
If a=1 then <-- color 2

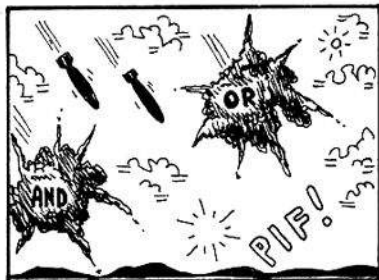
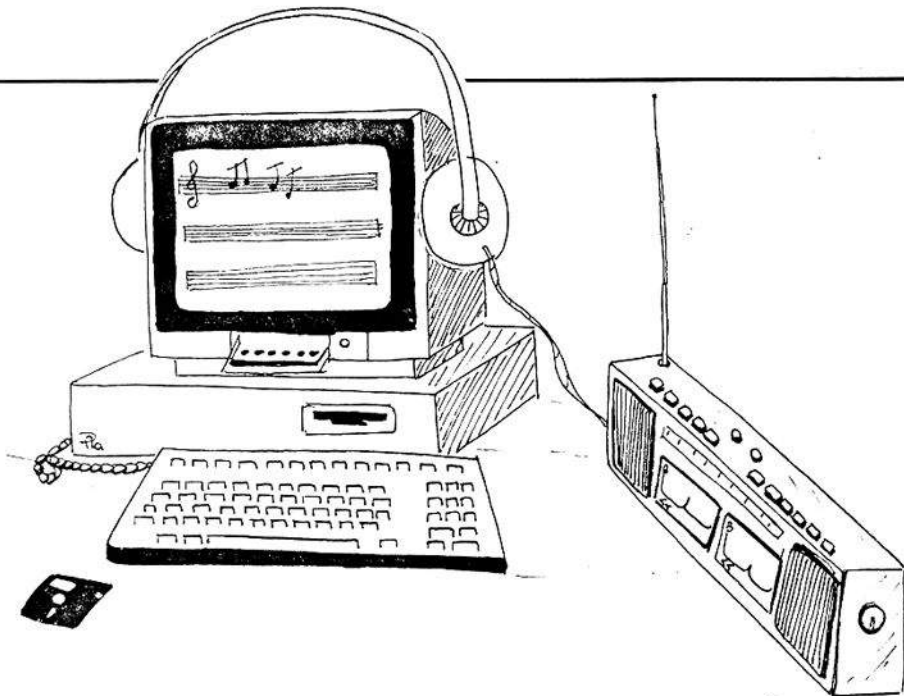
...pena l'emissione di un Syntax error.

Poichè il programma Basic Expander, dopo aver creato la tabella e letto la routine l.m, le scrive in memoria a partire dagli indirizzi specificati, bisognerà prestare particolare attenzione al posizionamento di questi indirizzi, in modo che il l.m. non vada a finire sopra la pagina 0, o sopra lo stesso programma Basic, o in altre zone altrettanto pericolose.

Bisognerà prestare anche attenzione alle zone di memoria che eventualmente usano le routine che svolgono i nuovi comandi, dato che queste potrebbero essere in conflitto con quelle utilizzate dalla execute statement, o potrebbero sovrapporsi alla execute statement o alla tabella.

Come routine per i nuovi comandi potranno essere utilizzate TUTTE le routine dell' "Enciclopedia di routine l.m." pubblicata periodicamente su C.C.C, nonchè qualsiasi routine scritta dall'utente o da altri, purchè terminino con RTS.

```
10 REM ************************
20 REM * BASIC EXPANDER  V1.0 *
30 REM * DI MARIANI GIANCARLO *
40 REM *   PER COMMODORE 64   *
50 REM ************************
60 :
70 REM -- INDIRIZZI DI DEFAULT --
80 RI=832:REM - INDIRIZZO DI PARTENZA DELLA ROUTINE
90 RF=RI+123:REM - IND. FINE
95 TS=52992:REM $CF00: TABELLA (INIZIO)
97 IF=TS+255:REM -$CFFF:TABELLA (FINE)
100 :
110 DIM CM$(100),IN$(100)
120 NM=1
130 SP$="                                    "
300 :
500 PRINT CHR$(147)
510 PRINT"-- BASIC EXPANDER V1.0 BY MARIANI G. --"
520 PRINT:PRINT" 1- LEGGI TABELLA"
530 PRINT" 2- SALVA TABELLA"
540 PRINT" 3- MODIFICA TABELLA"
550 PRINT" 4- MODIFICA INDIRIZZI"
560 PRINT" 5- SALVA LM"
570 PRINT" 6- FINE":PRINT
580 PRINT" INIZIO ROUTINE :";RI
590 PRINT" FINE ROUTINE   :";RF
600 PRINT" INIZIO TABELLA :";TS
610 PRINT" FINE TABELLA   :";IF:PRINT:PRINT
620 INPUT" SCEGLI";S
630 IF S<1 OR S>6 THEN PRINT CHR$(145);CHR$(145):GOTO 620
649 IF S=6 THEN PRINT CHR$(147):PRINT" CIAO!":END
650 ON S GOSUB 1000,2000,3000,4000,5000
660 GOTO 500
999 REM --- LEGGI TABELLA ---
1000 PRINTCHR$(147)
1080 INPUT" NOME ";N$:
1090 IF LEN(N$)=0 OR LEN(N$)>12 THEN 1000
1100 N$=N$+".TAB"
1105 OPEN15,8,15
1110 OPEN1,8,0,N$+",S,R"
1111 INPUT#15,A$,S$,D$,F$:IFA$<>"00" THEN 1200
1120 INPUT#1,NM
1130 FOR K=1 TO NM:INPUT#1,CM$(K),IN$(K):NEXT K
1140 CLOSE1:INPUT#15,A$,S$,D$,F$:PRINT:PRINT" ";A$;S$;D$;F$
1142 PRINT:INPUT" PREMI RETURN";D$
1150 CLOSE15:RETURN
1200 CLOSE1:CLOSE15:PRINT
1210 PRINT" ";A$;S$;D$;F$
1220 GOTO 1142
1999 REM -- SALVA TABELLA --
```

```
2000 GOSUB 4500
2080 INPUT" NOME ";N$:
2090 IF LEN(N$)=0 OR LEN(N$)>12 THEN PRINT CHR$(145);CHR$(145):GOTO 2080
2100 N$=N$+".TAB"
2105 OPEN15,8,15
2110 OPEN1,8,1,N$+",S,W"
2120 PRINT#1,(N2-N1+1)
2130 FOR P=N1 TO N2:PRINT#1,CM$(P):PRINT#1,IN$(P):NEXT P:PRINT#1
2140 CLOSE1:INPUT#15,A$,S$,D$,F$:PRINT:PRINT" ";A$;S$;D$;F$
2142 PRINT:INPUT" PREMI RETURN";D$
2150 CLOSE15:RETURN
2999 REM --- MODIFICA TABELLA ---
3000 GOSUB 4500
3020 IF N2>N1+20 THEN 3000
3070 PRINT CHR$(147)
3080 FOR K=N1 TO N2:K$=STR$(K)
3090 PRINTRIGHT$(SP$+K$,3);"- ";LEFT$(CM$(K)+SP$,20);"↑I- ";LEFT$(IN$(K)+SP$,10)
3500 NEXT K
3520 FOR K=N1 TO N2:K$=STR$(K)
3550 POKE 214,23:POKE 211,0:SYS 58732
3560 PRINTRIGHT$(SP$+K$,3);"- ";LEFT$(CM$(K)+SP$,20);"↑I- ";LEFT$(IN$(K)+SP$,10)
3570 CM$="":POKE 214,23:POKE 211,3:SYS 58732:INPUT CM$
3575 IF LEFT$(CM$,1)="*" THEN RETURN
3577 IF LEFT$(CM$,1)="=" THEN 3700
3580 CM$(K)="":FOR O=1 TO LEN(CM$):A$=MID$(CM$,O,1)
3590 IF A$="↑" OR A$=CHR$(32) THEN 3610
3600 CM$(K)=CM$(K)+A$:NEXT O
3610 POKE 214,23:POKE 211,27:SYS 58732:INPUT IN$(K)
3611 POKE 214,K-N1+1:POKE 211,0:SYS 58732
3615 PRINTRIGHT$(SP$+K$,3);"- ";LEFT$(CM$(K)+SP$,20);"↑I- ";LEFT$(IN$(K)+SP$,10)
3620 NEXT K:RETURN
3699 REM - DELETE -
3700 FOR L=K TO N2-1:CM$(L)=CM$(L+1):IN$(L)=IN$(L+1):NEXT L
3705 CM$(L)="":IN$(L)=""
3710 IF NM>1 THEN NM=NM-1
3715 IF N2>1 THEN N2=N2-1
3720 GOTO 3070
3999 REM --- MODIFICA INDIRIZZI ---
4000 POKE 214,10:POKE 211,16:SYS 58732:INPUT RI
4010 RF=RI+123:POKE 214,11:POKE 211,17:SYS 58732:PRINT "          "
4015 POKE 214,11:POKE 211,17:SYS 58732:PRINT RF
4020 POKE 214,12:POKE 211,16:SYS 58732:INPUT TS
4025 TF=TS+255:RETURN
4499 REM - INPUT N1,N2 -
4500 PRINTCHR$(147)
4510 PRINT" DAL NUMERO : 1"
4520 PRINT" AL NUMERO  :";NM
4530 PRINT CHR$(19):FOR H=1 TO 12:PRINT CHR$(29);:NEXT H:INPUT N1
4540 IF N1<1 OR N1>100 THEN 4500
4550 FOR H=1 TO 12:PRINT CHR$(29);:NEXT H:INPUT N2
```

```
4560 IF N2<1 OR N2>100 OR N2<N1 THEN 4500
4565 IF N2>NM THEN NM=N2
4570 PRINT:RETURN
4999 REM -- SALVA LM --
5000 GOSUB 4500
5005 REM - FORMA TABELLA -
5010 T$="":LT=0:FOR X=N1 TO N2
5020 L1=LEN(CM$(X)):L2=LEN(IN$(X))
5030 IF L1=0 OR L2=0 OR VAL(IN$(X))=0 OR VAL(IN$(X))>65535 THEN 5900
5040 FOR U=1 TO L1:U$=MID$(CM$(X),U,1)
5050 IF U$<"A" OR U$>"Z" THEN T$=LEFT$(T$,LEN(T$)-U+1):LT=LT-U+1:GOTO 5900
5060 LT=LT+1:IF LT>252 THEN 5850
5070 IF U=L1 THEN U$=CHR$(ASC(U$)+128)
5080 T$=T$+U$
5090 NEXT U
5100 IN=VAL(IN$(X)):I1=INT(IN/256):I2=IN-I1*256
5110 T$=T$+CHR$(I2)+CHR$(I1):LT=LT+2
5120 NEXT X
5130 T$=T$+CHR$(0):LT=LT+1
5140 FOR V=1 TO LT:POKE TS-1+V,ASC(MID$(T$,V,1)):NEXT
5150 PRINT:GOSUB 10000
5160 CK=0:FOR T=0 TO 26:READ A:POKE 679+T,A:CK=CK+A:NEXT T
5170 IF CK<>3440 THEN PRINT"ERRORE NEI DATI ROUTINE SAVE":END
5180 INPUT" NOME :";N$:IF LEN(N$)>9 THEN PRINT CHR$(145);CHR$(145):GOTO 5180
5185 RI$=RIGHT$(STR$(RI),LEN(STR$(RI))-1)
5190 TS$=RIGHT$(STR$(TS),LEN(STR$(TS))-1)
5200 N1$=N$+".R"+RI$:I1=INT(RI/256):I2=RI-I1*256
5210 U1=INT((RF+1)/256):U2=(RF+1)-U1*256
5220 POKE 251,I2:POKE 252,I1:POKE 700,U2:POKE 702,U1:POKE 689,LEN(N1$)
5225 FOR K=1 TO LEN(N1$):POKE 705+K,ASC(MID$(N1$,K,1)):NEXT K:SYS 679
5227 OPEN14,8,15:INPUT#14,A$,S$,D$,F$:CLOSE14
5228 PRINT:PRINT" R: ";A$;S$;D$;F$:PRINT
5230 N1$=N$+".T"+TS$:I1=INT(TS/256):I2=TS-I1*256
5240 U1=INT((TS+LT+1)/256):U2=(TS+LT+1)-U1*256
5250 POKE 251,I2:POKE 252,I1:POKE 700,U2:POKE 702,U1:POKE 689,LEN(N1$)
5260 FOR K=1 TO LEN(N1$):POKE 705+K,ASC(MID$(N1$,K,1)):NEXT K:SYS 679
5270 OPEN14,8,15:INPUT#14,A$,S$,D$,F$:CLOSE14
5280 PRINT" T: ";A$;S$;D$;F$:PRINT:INPUT" PREMI RETURN ";A$:RETURN
5799 END
5850 PRINT"ERRORE: TABELLA TROPPO LUNGA":PRINT
5860 INPUT"PREMI RETURN ";A$
5870 RETURN
5900 PRINT"COM.NON INSERITO:";CM$(X);"  ";IN$(X)
5910 GOTO 5120
10000 REM -- LETTURA DATI LM --
10010 RESTORE:CK=0:FOR K=0 TO 122:READ A
10020 CK=CK+A:POKE RI+K,A:NEXT K
10030 IF CK<>16608 THEN PRINT CHR$(147):PRINT"ERRORE NEI DATI ROUTINE LM!!":END
10040 N=40:GOSUB 11000
10050 N=50:GOSUB 11000
10060 N=58:GOSUB 11000
```

```
10070 N=81:GOSUB 11000
10080 N=101:GOSUB 11000
10090 N=107:GOSUB 11000
10100 R1=INT((RI+11)/256):R2=(RI+11)-R1*256
10110 POKE RI+1,R2:POKE RI+6,R1
10120 RETURN
11000 REM -- MODIFICA ROUTINE --
11010 T1=INT(TS/256):T2=TS-T1*256
11020 POKE N+RI,T2:POKE N+RI+1,T1
11030 RETURN
11097 :
11098 REM -- DATI ROUTINE LM --
11099 :
11100 DATA 169,000,141,008,003,169,000,141,009,003,096,165,122,133,250,165,123
11110 DATA 133,251,032,115,000,201,095,240,011,165,250,133,122,165,251,133,123
11120 DATA 076,228,167,162,000,189,000,207,201,128,176,042,032,115,000,221,000
11130 DATA 207,208,003,232,208,238,189,000,207,232,201,128,144,248,202,232,165
11140 DATA 250,133,122,165,251,133,123,232,232,032,115,000,189,000,207,208,210
11150 DATA 076,228,167,041,127,133,002,032,115,000,197,002,208,223,232,189,000
11160 DATA 207,133,250,232,189,000,207,133,251,032,115,000,169,167,072,169,173
11170 DATA 072,108,250,000
15000 REM ** DATI ROUTINE SAVE **
15100 DATA 169,015,162,008,160,001,032,186,255,169,000,162,194,160,002,032,189
15110 DATA 255,169,251,162,000,160,000,076,216,255
15120 END
```

```
Disassemblato di BASIC EXPANDER
By G. Mariani
(Rilocato da C000)

Routine che setta i puntatori

c000 lda #$0b      ;Setta i puntatori per
c002 sta $0308     ;la nuova routine di
c005 lda #$c0      ;Execute statement.
c007 sta $0309     ;
c00a rts           ;

Inizio nuova Execute Statement

c00b lda $7a       ;Salva i puntatori
c00d sta $fa       ;del Basic in $FA/$FB.
c00f lda $7b       ;
c011 sta $fb       ;
c013 jsr $0073     ;Prende caratt.succes.
c016 cmp #$5f      ;E' una freccetta?
c018 beq $c025     ;Si: Salta avanti
c01a lda $fa       ;No: Rimette a posto i
c01c sta $7a       ;puntatori del Basic e
c01e lda $fb       ;Salta alla normale
c020 sta $7b       ;routine S.O. di
c022 jmp $a7e4     ;Execute Statement.
c025 ldx #$00      ;Prende il primo car.
c027 lda $cf00,x   ;dalla tabella.
c02a cmp #$80      ;e' > di 128?
c02c bcs $c058     ;Si: salta avanti.
c02e jsr $0073     ;pross.carat dal basic
c031 cmp $cf00,x   ;Confronta con tabella
c034 bne $c039     ;Se diversi, salta.
c036 inx           ;Se no: increm. puntat
c037 bne $c027     ;Torna a confrontare.
c039 lda $cf00,x   ;Carattere da tab.
c03c inx           ;Incrementa puntatore.
c03d cmp #$80      ;Carattere > 128?
c03f bcc $c039     ;No: Torna a leggere.
c041 dex           ;
c042 inx           ;
c043 lda $fa       ;Rimetti a posto i
c045 sta $7a       ;puntatori del basic.
```

```
c047 lda $fb       ;
c049 sta $7b       ;
c04b inx           ;Incrementa il punt.
c04c inx           ;due volte.
c04d jsr $0073     ;Pross.carat.dal basic
c050 lda $cf00,x ;Carica carat. da tab.
c053 bne $c027     ;Se non e' 0, torna.
c055 jmp $a7e4     ;Se si: salta al S.O.
c058 and #$7f      ;Car=Car-128.
c05a sta $02       ;Salva car. in $02.
c05c jsr $0073     ;Legge pros.car. basic
c05f cmp $02       ;Confronta con $02.
c061 bne $c042     ;Se diversi, salta.
c063 inx           ;Se no: increm. punt.
c064 lda $cf00,x ;Leggi indirizzo LO.
c067 sta $fa       ;Salva in $FA.
c069 inx           ;Incrementa puntatore.
c06a lda $cf00,x ;Leggi indirizzo HI.
c06d sta $fb       ;Salva in $FB.
c06f jsr $0073     ;Pross.carat.dal basic
c072 lda #$a7      ;predisponi l'indiriz.
c074 pha           ;di ritorno a $A7AE
c075 lda #$ad      ;(Esegui prossimo
c077 pha           ; comando basic)
c078 jmp ($00fa) ;Salta a user routine.

Routine di SAVE by Mariani G.
(Rilocata da $0340)

0340 lda #$0f      ;Numero file (15).
0342 ldx #$08      ;periferica (8).
0344 ldy #$01      ;Indirizzo secondario.
0346 jsr $ffba     ;Setta i parametri.
0349 lda #$00      ;Lunghezza nome.
034b ldx #$c2      ;Ind. basso Nome.
034d ldy #$02      ;ind. alto Nome.
034f jsr $ffbd     ;Setta param. per nome
0352 lda #$fb      ;Ind. partenza (FB/FC)
0354 ldx #$00      ;Indirizzo di fine
0356 ldy #$00      ;registrazione (X/Y).
0358 jmp $ffd8     ;Routine S.O. di SAVE
```

MA CHE BELLE ROUTINE

Come utilizzare alcune potenti routine l.m. presenti nelle Rom del vostro C/64

di **Giancarlo Mariani**

Il C/64, come tutti i piccoli computer della famiglia Commodore (VIC/20, C/16, C/128) possiedono il sistema operativo in una ROM, che evita di doverlo caricare tutte le volte che si accende la macchina, come invece succede operando con gli IBM e con i professionali in genere (tra cui l'Amiga).

Il sistema operativo (S.O.) contiene tutte le routine che servono a far funzionare correttamente l'elaboratore. Gran parte di queste sono sfruttabili dall'utente nello sviluppo di programmi Basic o l.m. con un conseguente risparmio di notevole lavoro di programmazione.

Nel presente articolo ne mostreremo alcune, utili per molteplici scopi, accompagnate dal modo di utilizzo, per meglio comprenderne la funzione.

$AEFD: Controlla la presenza di una virgola posta dopo un comando Basic. Utilizzo:
- Saltare alla routine

$AEF7: Controlla la presenza di una parentesi chiusa

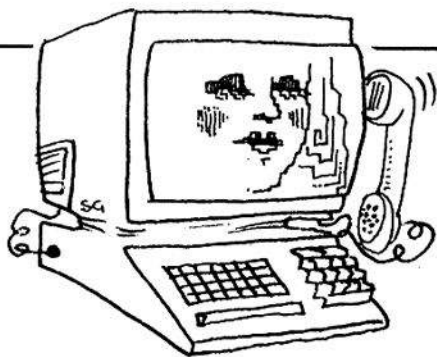
$AEFA: Controllo parentesi aperta

$AEFF: Controlla la presenza di un carattere qualsiasi dopo un comando Basic. Il carattere deve essere contenuto nell'accumulatore. Utilizzo:
- Caricare in A il carattere
- Saltare alla routine

$B79E: Preleva un numero, compreso tra 0 e 255, presente dopo un comando Basic e lo mette nel registro X. Utilizzo:
- Saltare alla routine
- Utilizzare il numero in X

$AD8A / $B7F7: Prelevano un numero, compreso tra 0 e 65535, dopo un comando Basic e lo mettono, rispettivamente, in $14 (byte basso) e $15 (byte alto). Utilizzo:
- Saltare a $AD8A
- Saltare a $B7F7
- Leggere il numero da $14 / $15

$AD9E / $B782: Prelevano una stringa

E' inutile perdere tempo a programmare determinate routine se queste sono già disponibili su Rom

__Utilizzare la routine di una Rom è semplice: basta sapere l'indirizzo di partenza e le modalità di attivazione__

presente dopo un comando Basic. La lunghezza della stringa viene messa in Y, mentre la stringa è puntata da $22 / $23. Utilizzo:
- Saltare a $AD9E
- Saltare a $B782
- Utilizzare Y / $22 / $23 per leggere la stringa

$AD9E: Valuta l'espressione oppure preleva un parametro reale o stringa presente dopo un comando Basic. Nella locazione $0D ci sarà 0 se l'espressione è numerica, oppure 255 se è stringa. Utilizzo:
- Saltare alla routine
- Leggere $0D oppure leggere il numero reale dall'accumulatore in virgola mobile n. 1

$B113: Controlla il carattere in A. Se è numerico il carry verrà messo a 0, mentre se è alfabetico il carry sarà a 1. Utilizzo:
- Caricare il carattere nell'accumulatore
- Saltare alla routine
- Leggere il carry

$AB1E: Stampa una stringa. Il byte basso di inizio stringa deve essere in A, il byte alto in Y. Alla fine della stringa deve essere presente uno 0. Utilizzo:
- Caricare in A il byte basso di inizio stringa
- Caricare in Y il byte alto
- Saltare alla routine

$AB24: Stampa una stringa puntata da $22 / $23. La lunghezza deve essere in A. Utilizzo:
- Caricare in $22 il byte basso dell'inizio della stringa
- Caricare in $23 il byte alto
- Caricare nell'accumulatore la lunghezza della stringa
- Saltare alla routine

$AB25: Stessa cosa di $AB24, solo che la lunghezza della stringa deve essere nel registro X

$FFD2: Manda sul canale di uscita (normalmente il video) il carattere contenuto in A. Utilizzo:
- Caricare nell'accumulatore il carattere da inviare
- Saltare alla routine

$BDCD: Stampa un numero compreso tra 0 e 65535. Il byte basso deve essere in X mentre quello alto in A. Utilizzo:
- Caricare in X il byte basso del numero da stampare
- Caricare nell'accumulatore il byte alto
- Saltare alla routine

$AABC: Stampa un numero in virgola mobile contenuto nel FLP ACC#1. Utilizzo:
- Saltare alla routine

$E38B: Stampa un messaggio di errore, il cui codice deve essere contenuto in X, secondo la seguente tabella:

$01: Too many files
$02: File open
$03: File not open
$04: File not found
$05: Device not present
$06: Not input file
$07: Not output file
$08: Missing file name
$09: Illegal device number
$0A: Next without for
$0B: Syntax
$0C: Return without gosub
$0D: Out of data
$0E: Illegal quantity
$0F: Overflow
$10: Out of memory
$11: Undef'd statement
$12: Bad subscript
$13: Redim'd array
$14: Division by zero
$15: Illegal direct
$16: Type mismatch
$17: String too long
$18: File data
$19: Formula too complex
$1A: Can't continue
$1B: Undef'd function
$1C: Verify
$1D: Load
$1E: Break

Utilizzo:
- Caricare in X il codice dell'errore
- Saltare alla routine

$FFBA: Fissa il numero di file logico, il numero di periferica e l'indirizzo secondario nel trattamento dei file. Utilizzo:

- Caricare il numero del file in A
- Caricare il numero di periferica in X, secondo la seguente tabella:

0: Tastiera
1: Registratore
2: RS-232
3: Schermo
4: Stampante
8: Disco

- Caricare l'indirizzo secondario in Y
- Saltare alla routine

$FFBD: Fissa il nome del file. Utilizzo:
- Caricare in A la lunghezza del nome
- Caricare in X la parte bassa dell'indirizzo di partenza del nome
- Caricare in Y la parte alta
- Saltare alla routine

$FFC0: Apre un file. Utilizzo:
- Utilizzare la $FFBA
- Utilizzare la $FFBD
- Saltare alla routine

$FFD5: Carica un programma o un file da periferica. Utilizzo:

- Utilizzare la $FFBA
- Utilizzare la $FFBD
- Mettere A a 0 per LOAD oppure a 1 per VERIFY.
- Nel caso in cui l'indirizzo secondario sia 0, caricare in X la parte bassa dell'indirizzo di partenza per Load ed in Y la parte alta
- Saltare alla routine

$FFD8: Salva la ram su periferica. Utilizzo:
- Utilizzare la $FFBA
- Utilizzare la $FFBD
- Mettere in due locazioni di memoria consecutive di pagina 0 l'indirizzo di partenza del pezzo di memoria da salvare nella forma byte basso - byte alto.
- Caricare nell'accumulatore la prima di queste due locazioni di memoria
- Caricare in X il byte basso della fine della memoria da salvare
- Caricare in Y il byte alto
- Saltare alla routine

$FFC6: Predispone nel modo di input un file già aperto. Utilizzo:
- Aprire il file
- Caricare in X il numero del file
- Saltare alla routine

$FFC9: Predispone nel modo di output un file già aperto. Utilizzo:
- Aprire il file
- Caricare in X il numero del file
- Saltare alla routine.

$FFCF: Legge un carattere da un canale di input e lo mette in A. Utilizzo:
- Aprire il canale
- Utilizzare la $FFC6
- Saltare alla routine
- Utilizzare il dato in A

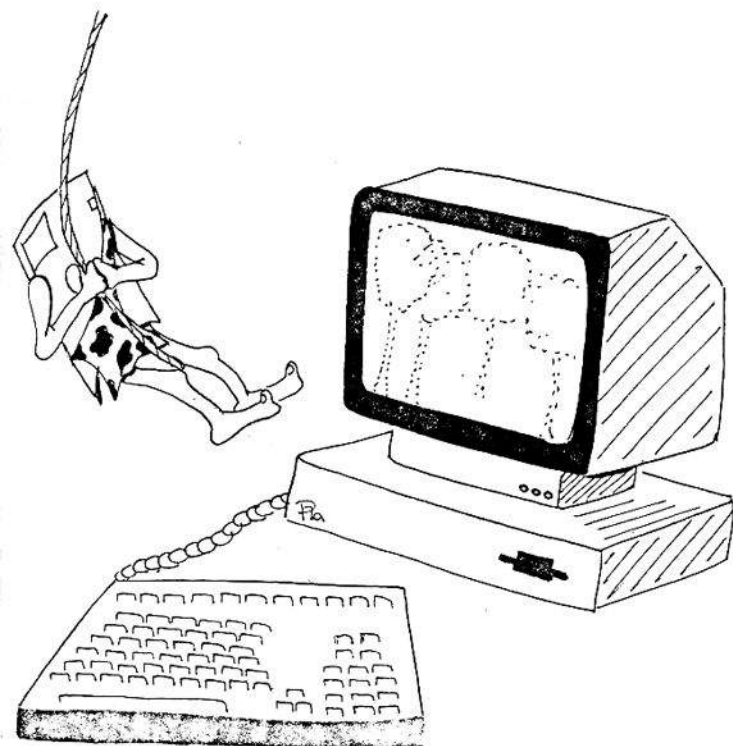
NB: Se il canale di ingresso è la tastiera si possono saltare i primi due passi

$FFD2: Manda in uscita un carattere su un canale già aperto. Utilizzo:
- Aprire il canale
- Utilizzare la $FFC9
- Caricare in A il dato da inviare
- Saltare alla routine

NB: Se il canale in uscita è il video si possono saltare i primi due passi.

$FFCC: Resetta tutti i canali in I / O. La periferica di default in input ridiventa la tastiera mentre in output il video. Utilizzo:
- Saltare alla routine

$FFC3: Chiude un file logico. Utilizzo:
- Caricare in A il numero del file da chiudere
- Saltare alla routine

A volte è sufficiente la sola Sys di partenza per disporre di una funzione; altre volte, invece, è necessario settare alcuni registri interni del microprocessore

Qualsiasi computer contiene (almeno) una Rom; le routine ivi contenute facilitano la vita del programmatore

$FFE7: Chiude tutti i file attualmente aperti. Utilizzo:
- Saltare alla routine

$FFB7: Mette nell'accumulatore il valore dello status register (ST), secondo la tabella:

$01: Time out in scrittura
$02: Time out in lettura
$04: Blocco corto
$08: Blocco lungo
$10: Errore irrecuperabile in lettura
$20: Errore di checksum
$40: End of file
$80: End of tape (sul bus seriale significa device not present)

Utilizzo:
- Saltare alla routine
- Leggere il valore di A

$EB48: Scandisce la tastiera e mette il valore Ascii del tasto premuto nel registro X. Utilizzo:
- Saltare alla routine
- Leggere, in X, il codice Ascii del tasto premuto

$A68E: Esegue un reset del Basic e fa partire il programma presente in memoria. Utilizzo:
- Saltare alla routine

$FF9C: legge o fissa la parte più bassa della memoria. Utilizzo:
Per leggere il punto di partenza della memoria Ram:

- Mettere il carry a 1
- Saltare alla routine
- Leggere il byte basso di partenza della Ram dal registro X
- Leggere il byte alto da Y

Per fissare il limite basso della memoria Ram:
- Mettere il carry a 0
- Mattere in X il byte basso di partenza della memoria Ram
- Mettere in Y il byte alto
- Saltare alla routine

$FF99: legge o fissa il punto massimo (più alto) di memoria Ram. Utilizzo: stesso di $FF9C.

$FFF0: Legge o fissa la posizione del cursore. Utilzzo:
Per leggere la posizione:
- Mettere il carry a 1
- Saltare alla routine
- Leggere da X la riga e da Y la colonna della posizione del cursore

Per posizionare il cursore:
- Mettere il carry a 0
- Caricare in X la riga desiderata
- Caricare in Y la colonna
- Saltare alla routine

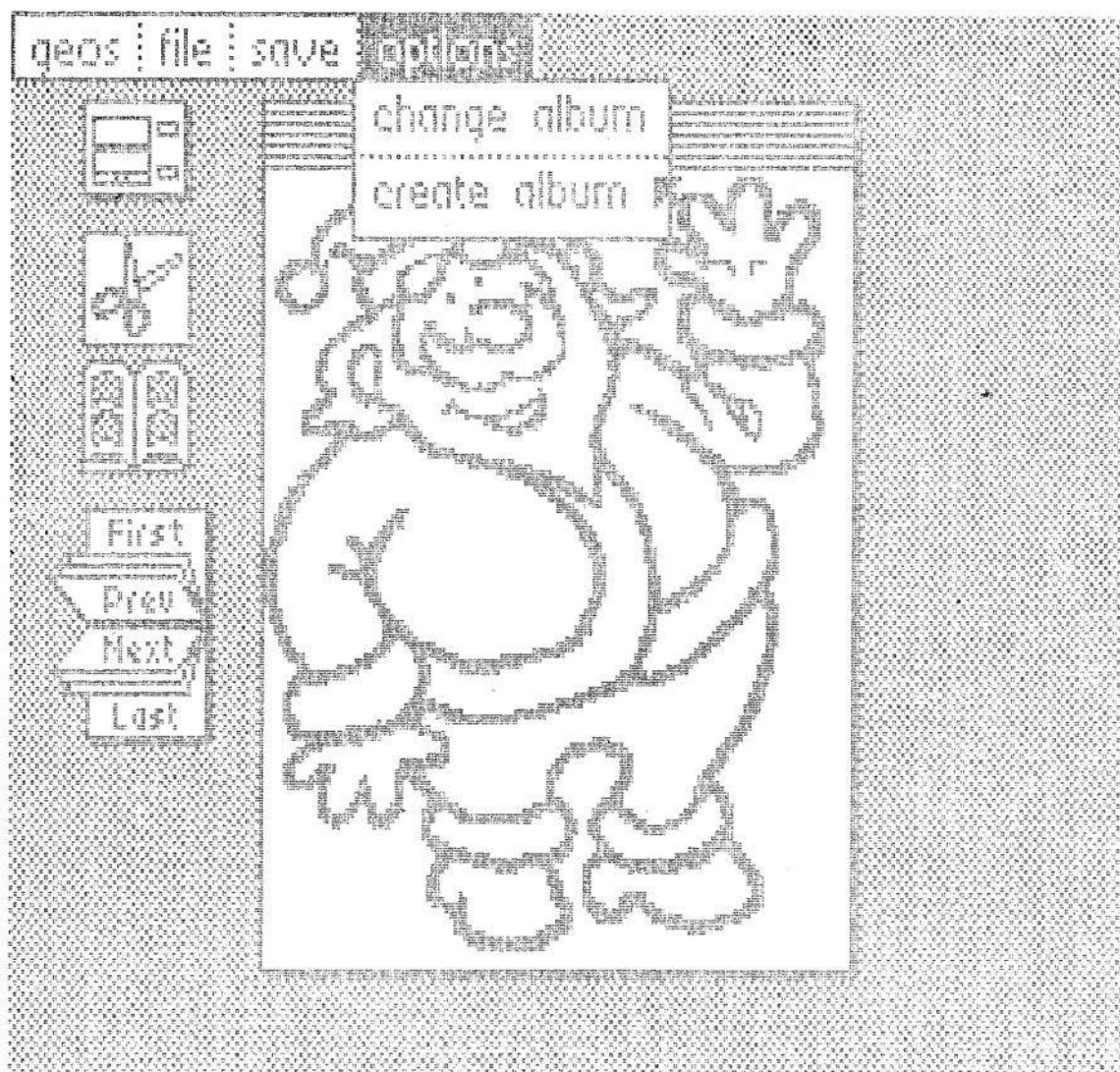
Con questa prima carrellata di routine termina l'articolo. Se si dovesse riscontrare un interessamento da parte dei lettori, ne verranno pubblicate altre, oppure chiarimenti su quelle presentate.

Un'ultima nota, prima di concludere, riguarda la lettera di un lettore che chiedeva spiegazioni sulla routine posta all'indirizzo $A4A4. Ebbene, questa routine ricerca una linea, in un programma Basic, il cui numero deve essere contenuto nelle locazioni $14/$15; se risulta presente, la cancella, altrimenti la inserisce nel programma stesso. Il modo di utilizzo è il seguente:
- Caricare in $14 il byte basso del numero di linea
- Caricare in $15 il byte alto
- Saltare alla routine $A4A4

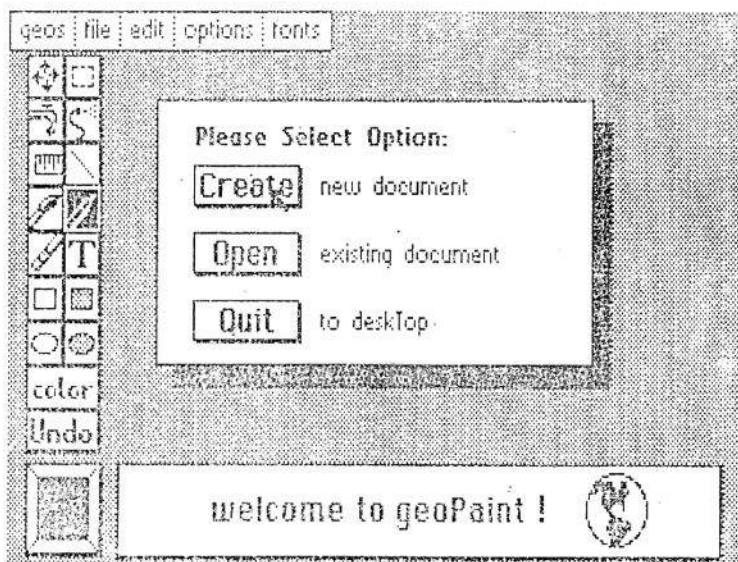
VR. VIDEOREGISTRARE

Fra qualche anno suoni, dati e immagini utilizzeranno gli stessi supporti. Il televisore diventerà un po' computer, così come già oggi il computer è uno strumento di svago e di evasione ed entrambi avranno in comune le stesse risorse di memorizzazione (videotape, disco laser digitale) e di input "visive" come le telecamere, soprattutto per le applicazioni robotiche.

Tecnicamente tutto ciò è già possibile collegando il computer col videoregistratore, la cui diffusione entro il 1990 dovrà raggiungere il 60% delle famiglie.
VR-Videoregistrare, è la rivista SYSTEMS rivolta agli utenti che vogliono fare un passo avanti nell'utilizzazione del videotape e della telecamera.

MS-DOS
&
GW-BASIC
EMULATOR 2.0
S systems

GEOS, L'ALTRO MODO DI DIRE C/64

Se hai appena comprato un drive, forse vale la pena saperne di più su un nuovo sistema operativo per C/64

di **Antonio Pastorelli**

Il C/64 è stato, ed è tuttora, il computer che ha segnato un profondo solco nella storia dell'informatica.

Grazie al C/64, infatti, moltissimi si sono avvicinati al mondo dell'informatica per motivi di studio, lavoro o puro divertimento.

Ma da che cosa deriva tanto successo per una macchina che, in fondo, è dotata di limitate risorse?

In buona parte ha contribuito sicuramente l'apertura del sistema a qualsiasi manomissione da parte degli utenti e dei programmatori; sono ormai numerosi i ragazzi che, grazie al C/64, si sono trasformati in veri e propri "mostri", gettando le basi per affrontare con sicurezza la sistemi più sofisticati e professionali.

Questa caratteristica fondamentale del C/64 è stata pienamente sfruttata dalla Berkeley Softworks nel presentare il Geos.

Il Geos è un vero e proprio sistema operativo (S.O.) che si sostituisce a quello presente su ROM al momento dell'accensione del calcolatore.

Grazie al Geos, il C/64 si trasforma in un computer completamente diverso: viene ottimizzata la gestione del drive, l'interfaccia utente, le capacità grafiche, ed in generale il modo di operare del computer.

E' importante sottolineare che per il suo corretto funzionamento è necessario possedere un C/64 (oppure un C-128), un drive ed un joystick.

IL DISCO

La gestione del disco è completamente rinnovata sia per ciò che riguarda la velocità delle operazioni di Input/Output, che la memorizzazione dei file.

Le operazioni di salvataggio e caricamento, ad essi relative, risultano essere, infatti, di quasi 6 volte più veloci delle corrispondenti operazioni eseguite in modo standard.

Tali prestazioni sono quelle che normalmente si ottengono con programmi velocizzatori, o tramite alcune cartucce.

Tuttavia, in alcuni casi, il vantaggio di questi ultimi sistemi viene a mancare, dal momento che non sono compatibili con tutti i programmi in circolazione; inoltre velocizzano solo il caricamento dei file di programma e si disattivano quando, caricato un programma, lo si manda in esecuzione.

Se, ad esempio, carichiamo un programma di word processing, i testi battuti verranno registrati su disco a velocità normale, mentre, con Geos, il turbo è sempre attivo e velocizza tutte le operazioni con il disco.

Per ogni file presente sul disco, il Geos memorizza una scheda informativa contenente il nome del disco stesso, il tipo ed il nome del file, la sua lunghezza (in Kbyte), la data e l'ora dell'ultima modifica, l'autore, l'eventuale protezione da scrittura, e perfino note aggiuntive.

Queste preziose informazioni, che al giudizio di un profano sembrerebbero scontate, sono una vera e propria scoperta per l'utente di C/64 abituato a registrare i file con la sola indicazione del nome, lungo al massimo 16 caratteri.

La possibilità di introdurre altre note, tra le informazioni relative ai file, evita di dimenticare, a distanza di tempo, il loro contenuto, con la conseguente involontaria cancellazione.

L'INTERFACCIA UTENTE

Il sistema di comunicazione, tra computer e utente, non è più il solito rigido "editor", cioè un colloquio basato sull'utilizzo di parole-chiave incaricate di formare frasi-comando da interpretare ed eseguire successivamente; tutto viene deciso tramite l'uso della leva del joystick e confermato con il pulsante di sparo.

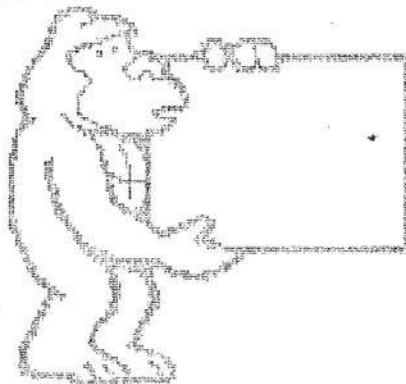
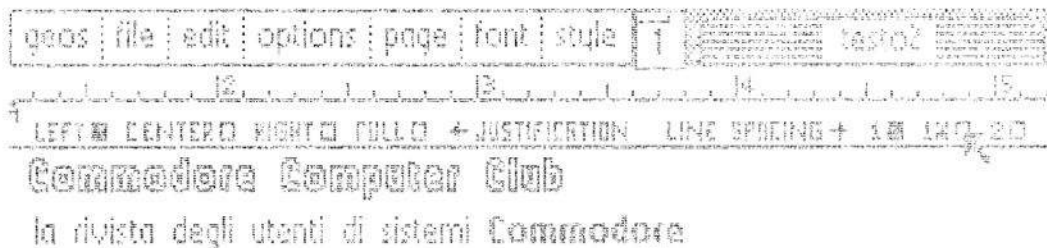
In tal modo, caricare un file, eseguire un copia, cambiare nome ad un disco, leggere una directory, diventano operazioni di una semplicità estrema: si muove la leva ed il gioco è fatto!

Questo modo di comunicare con il calcolatore è stato utilizzato anche su altri computer, ben più potenti, come Amiga e Atari ST.

Su Amiga è stato battezzato "Intuition", dal momento che tutte le operazioni di colloquio tra uomo e mac-

geos | file | edit | options | page | font | style

Commodore Computer Club

la rivista degli utenti di sistemi Commodore

Questo e' un esempio di testo con grafica, prelevata da Newsroom con il 'Graphic Grabber' (Disco Deskpack).

china avvengono attraverso la selezione di menu ed immagini, che ricordano, intuitivamente (appunto), la funzione desiderata.

Sembra proprio che i tempi in cui occorreva laurearsi per utilizzare il computer siano proprio finiti! Lavorando con Geos non è più necessario memorizzare centinaia di istruzioni, o regole sintattiche, per l'introduzione manuale dei comandi; tutto ciò che occorre lo si reperisce su video e lo si seleziona con la pressione di un solo tasto.

I PROGRAMMI

Il Geos viene fornito con alcuni programmi che ne permettono l'immediato utilizzo, almeno per quanto riguarda le applicazioni più frequenti.

Sul disco del sistema operativo (quello dal quale viene attivato il Geos), si trovano anche "Geowrite" e "Geopaint", due ottimi programmi di word-processing (elaborazione di testi) e di grafica.

Grazie all'architettura del nuovo S.O. è possibile avere costantemente disponibili dei tools che non interferiscono minimamente con il programma che si sta eseguendo.

Mentre, ad esempio, disegnate con Geopaint, potete richiamare un taccuino, sul quale memorizzare alcuni appunti, per poi ritornare al disegno lasciato in sospeso.

Il meccanismo che permette di lavorare contemporaneamente su diversi fronti, viene chiamato multitasking ed è una caratteristica tipica di computer molto evoluti.

Chiaramente su Commodore 64 non è pensabile un vero e proprio multitasking, ma questo viene simulato, in maniera soddisfacente, dal Geos.

D'altro canto, un computer che disponga di un vero e proprio multitasking costerebbe svariati milioni.

Tra i tools a disposizione vi sono la CALCOLATRICE, l'OROLOGIO con sveglia, la data ed ora, il taccuino, il Photo-manager e le PREFERENZE.

Il Photo-manager è una utility che facilita la manipolazione e l'archiviazione di immagini in alta risoluzione, per la successiva utilizzazione in programmi applicativi (ad esempio, in Geowrite).

In ogni caso i programmi in Basic, scritti "sotto" sistema operativo standard, vengono riconosciuti e trattati come un qualsiasi altro programma eseguibile da Geos.

Se avete acquistato l'espansione di memoria, e non avete ancora trovato un programma che ne faccia uso, questo è il package giusto per sfruttarla; il Geos è in grado di utilizzarla correttamente.

PROSSIMAMENTE

La Berkeley pubblicherà un manuale del Geos, destinato ai programmatori, per facilitare l'utilizzo delle potenti routine disponibili; sembra, inoltre, che presenterà un interprete Basic evoluto, in grado di trarre vantaggio dalle potenzialità del Geos, che, altrimenti, rimarrebbero nascoste.

Oltre ai programmi presenti sul disco Geos, esistono numerosi programmi applicativi, i quali permettono di ampliare le possibilità operative della confezione di base.

Daremo di seguito uno sguardo a "Deskpack" ed a "Geowrite workshop".

DESKPACK

Si tratta di un package composto da sette programmi di utilità, vitale per chi lavora abitualmente con il Geos.

Innanzitutto viene migliorato il Desktop, con l'aggiunta di DRIVER per stampanti e per le periferiche di input.

Un driver è un file che contiene le istruzioni per il corretto utilizzo di un determinato dispositivo; il driver per la stampante "Commodore 803", quindi, è rappresentato dall'insieme di istruzioni in l.m. che il computer dovrà seguire, in modo totalmente automatico, per pilotare correttamente la stampante MPS-803.

GEODEX è un database per l'archiviazione, l'ordinamento e la stampa di indirizzi (anche su etichette, in diversi formati), collegabile al modem/1200 per la selezione automatica del numero; questo programma, unitamente a GEOMERGE, utility per la produzione di copie multiple di documenti di vario genere, permette una completa ed efficiente gestione di dati.

Con GRAPHIC GRABBER sarete in grado di riutilizzare le numerose immagini dei programmi di grafica più diffusi per il C/64, tra cui Printshop, Printmaster e Newsroom.

Dopo aver "catturato" le immagini, queste saranno memorizzate in un album, gestibile mediante PHOTO MANAGER, per il riutilizzo con gli applicativi Geos: GEOPAINT, GEOWRITE e GEOPUBLISH.

L'ICON EDITOR permette di modificare a piacimento le icone dei file (immagini assegnate agli archivi di dati o programmi, visualizzate nella finestra della directory di Desktop), e di adattare i file non-Geos al nuovo formato basato sulle icone.

Un CALENDARIO perpetuo, con AGENDA per appunti, è richiamabile in qualsiasi momento, anche da altri applicativi, ritornando a ciò che si stava eseguendo, senza essere costretti ad uscire da un programma per caricarli, come si aspetterebbe un comune utilizzatore di C/64.

Infine, per i momenti di distensione, non poteva mancare il classico gioco da tavolo, il BLACKJACK.

GEOWRITE WORKSHOP

Geowrite Workshop è un package contenente la versione 2.1 del programma Geowrite, già presente nel disco di sistema, ma dotato di altre utility, come il driver per la stampante laser(!), il Text Grabber, Geomerge, ed i Paint Drivers.

Le caratteristiche peculiari della nuova release di Geowrite sono le seguenti:

- formattazione separata di ciascun paragrafo
- giustificazione del testo a destra, centrata o completa
- spaziatura singola, doppia ed intermedia
- stampa di bozza, di qualità, Letter quality
- stampa parziale di un documento
- ricerca e sostituzione del testo

Con Geowrite è possibile scrivere documenti che comprendono grafici, creati con Geopaint, includere testi provenienti da altri documenti, o anche da altri word processor per C/64, tramite il "Text Grabber" che si occupa delle opportune conversioni.

Nello stesso disco si trovano "Geomerge", già descritto, un Desktop migliorato ed ampliato con nuovi driver per stampanti e per dispositivi di input, tra i quali anche GEOLASER.

Tramite Geolaser è possibile utilizzare una stampante laser, collegata al C/64 tramite la User Port (RS-232).

Facendo uso di un tale tipo di stampante si può accedere, oltre che ai normali font di caratteri messi a disposizione da Geowrite, anche a quelli propri della stampante, ottenendo una qualità di stampa quasi tipografica.

CONSIDERAZIONI

Tramite Geos è possibile pensare a soluzioni semi-professionali a basso costo, dal momento che basta un C/64 (o 128) un drive, una stampante ed un joystick per utilizzare tutte le possibilità offerte da una gamma di programmi piuttosto professionali, sia per la veste grafica che per quella operativa.

Tali programmi potrebbero essere paragonati a quelli disponibili per sistemi ben più evoluti (Ms-Dos), se non fosse per la modesta velocità del microprocessore (6510) e per la quantità di memoria, solo 64 KRam, per applicazioni che, in Ms-Dos, ne richiedono 640.

Il tutto è condito da manuali in italiano, esaurientemente esplicativi, che non lasciano alcun dubbio circa il corretto e completo utilizzo dei programmi.

I prodotti Geos sono distribuiti da:

Leader Distribuzione
Via Mazzini 14
20120 Casciago (VA)
Tel 0332/212255

SYSTEMS NEWS

Un bollettino mensile per tenervi informati sui più interessanti prodotti Systems

La quasi totalità dei prodotti Systems, come è noto, viene distribuita in edicola restando quindi disponibile per un tempo relativamente breve (uno o due mesi al massimo) in funzione della loro periodicità.

Ciò significa che, molto probabilmente, vi è sfuggito qualche fascicolo, disco o cassetta, che avrebbe potuto arricchire la vostra softwarteca.

Forse non tutti sanno che è sempre e comunque possibile (a meno che non sia esaurito) richiedere un qualsiasi prodotto al nostro ufficio arretrati che, con la sola aggiunta delle spese postali, provvederà a farvela recapitare direttamente a casa a stretto giro di posta.

Il presente bollettino ha il principale scopo di tenere costantemente informati i nostri lettori sulle ultime pubblicazioni Systems e, in generale, sui prodotti più interessanti, soprattutto per chi intende usare il computer in modo produttivo e didattico.

I prodotti Systems sono schematizzabili in due grandi categorie: riviste e software su supporto magnetico.

Alla prima categoria appartengono le pubblicazioni come "Commodore Computer Club", "Personal Computer", "Computer" e tutti i fascicoli "speciali" che trattano argomenti troppo vasti per essere divulgati sotto forma di semplici articoli sulle riviste.

Le cassette e i prodotti su disco appartengono, invece, alla seconda categoria.

Anche qui vi sono diversificazioni: si parte da "Software Club" (bimestrale su cassetta contenente diversi videogame ed utility), dedicato agli utenti più giovani, per arrivare a "Commodore 64 Club", pubblicazione anch'essa bimestrale, ma su disco.

Si tratta di un floppy, per C/64-128, registrato interamente su entrambe le facce (oltre 300 Kbyte di software) contenente (come, del resto, "Software Club") software italiano totalmente originale, dedicato, però, ad utenti di disk drive e contenente sia giochi sia programmi di utilità di ottimo livello.

Anche per la linea di prodotti software esiste una categoria, a parte, che racchiude i prodotti "speciali" in grado di offrire programmi o applicazioni di particolare interesse.

Le modalità di richiesta sono le seguenti:

- Compliare un modulo di C/C postale indirizzato a:

Systems Editoriale
C/C postale N. 37952207
V.le Famagosta, 75
20142 Milano

- Per ogni spedizione vanno aggiunte L. 3000 per spese postali (o L. 6000 per Raccomandata) oltre al prezzo di copertina del prodotto in questione.

Inutile dire che se si ordinano simultaneamente più prodotti, le spese postali verranno pagate una volta sola.

Per ulteriori informazioni si prega, tuttavia, di contattare direttamente la responsabile del servizio arretrati, Lucia Dominioni (tel. 02/84.67.348).

COMMODORE UTILITIES

Un intero dischetto per C/64 contenente ben 14 utility ed un super videogame.

Si tratta di interessanti programmi di utilità che si riveleranno più volte indispensabili per chi, esperto o meno, ama trafficare con il C/64.

Dal semplice Data Maker, che consente di trasformare una zona di memoria in linee DATA, al fantastico Graph Compiler che, tramite apposito linguaggio, consente applicazioni grafiche impressionanti.

Dal Turbokit, che permette di velocizzare fino a 5 volte il caricamento di ogni programma, all'Autorun Maker che permette di dotare di Autorun qualsiasi programma Basic senza difficoltà alcuna.

Particolare attenzione va posta alla serie di programmi per la gestione dei file, come Format Saver e Disk Manager, che consentono, rispettivamente, di recuperare il contenuto di un dischetto formattato per errore e di effettuare le più importanti operazioni di modifica sui file.

Prezzo: L. 15000 (solo su disco)

C 64 CLUB N. 6

Il gioco "di testa" è Work Games, un bellissimo software che, sullo stile di Summer e Winter Games, propone una serie di prove ambientate, però, in... fabbrica.

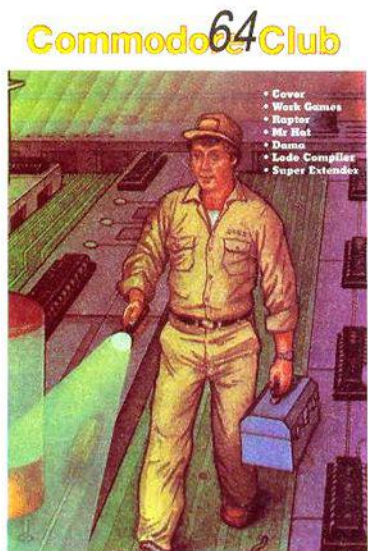
I giocatori (da uno a sei) impersoneranno altrettanti operai impegnati in numerose ed avvincenti prove.

Mr. Hat, è invece un gioco "a livelli" che vede il protagonista impegnato nella ricerca di un magico cappello d'oro.

Raptor è ambientato in un'epoca indefinita in cui un malvagio stregone ha rapito le più belle fanciulle per trasformarle in streghe. Il tuo compito? Salvarle, naturalmente!

Il quarto gioco è la classica "Dama", di ottimo livello, che permette una quantità di opzioni tra cui il gioco automatico e lo studio delle strategie.

Le due utility presenti sono destinate ad utenti esperti; permettono, rispettivamente, di creare comandi Basic personalizzati (Superextender) e di compilare propri programmi dotandoli, inoltre, di una protezione di tipo password.

Prezzo: L. 12000 (solo su disco)

SOFTWARE CLUB N. 19

Bimestrale su cassetta, questo numero contiene numerosi videogame più un'interssante utility per i possessori di registratore a cassette.

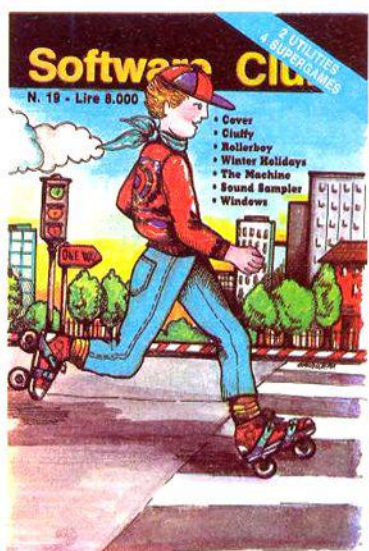
I giochi sono ben quattro e toccano i più disparati argomenti: Rollerboy vi farà vivere le emozionanti avventure di un teenager che, con i pattini ai piedi, pecorre le strade della città alla ricerca di oggetti e denaro.

Ciuffy, un simpatico cagnolino, vi chiede di essere guidato per sfuggire ai suoi numerosi nemici.

The machine, un classico gioco di automobili, vi vedrà impegnati in una gara ad altissima tensione (e velocità).

Winter Holidays vi divertirà facendovi impersonare un manovratore di funivie con qualche problema agli impianti...

Sound Sampler e Windows sono le due fantastiche utility che, rispettivamente, permettono di ascoltare, tramite computer, una qualsiasi cassetta musicale, e di gestire il C/64 tramite finestre per mezzo di appositi comandi.

Prezzo: L. 8000 (a richiesta, su disco, L. 12000)

COMPUTER MUSIC

E' un software su cassetta che contiene un programma per la creazione di musiche da inserire nei propri programmi Basic o L.M.

Contiene numerose motivi scritti dagli artisti più affermati, da Vasco Rossi a Madonna, dai Beatles ai Duran Duran.

La cassetta, non essendo più presente in edicola, può essere reperita esclusivamente presso il nostro ufficio arretrati.

Prezzo: L. 12000 (solo cassetta)

MS DOS - GW BASIC EMULATOR

E' la seconda versione, che ha superato di gran lunga il successo della precedente.

Un package di altissi mo livello che permette di utilizzare il C/64 in ambiente MS DOS, il Sistema Operativo dei computer professionali IBM compatibili.

Progettato e realizzato per fornire un potente strumento didattico a tutti coloro che, per motivi di studio o di lavoro utilizzano un Pc IBM (ma non lo possiedono), ha aperto gli orizzonti dell'informatica professionale a migliaia di studenti, sia in Italia che in altri paesi europei, dove è stato esportato.

Contiene un simulatore di MS DOS e un nuovo interprete Basic (GW Basic) oltre al manuale di istruzioni contenente numerosi esempi applicativi.

Prezzo: L. 19000 (disponibile solo su disco)

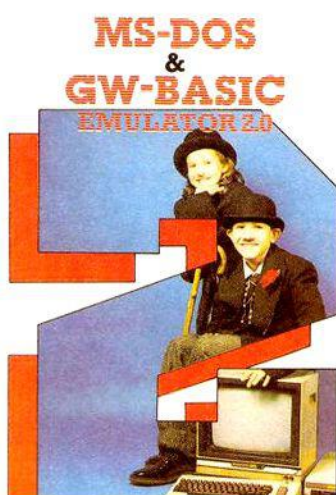
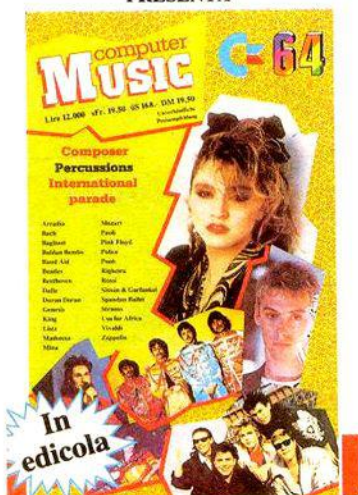
COMMODORE SPECIALE TOTOCALCIO

Un intero fascicolo di ben 116 pagine dedicato al Totocalcio che tratta, in modo completo, tutti gli argomenti ad esso inerenti.

Dai sistemi condizionati alle riduzioni, dai sistemi a correzione di errore al calcolo probabilistico applicato al popolare gioco nazionale.

Decine di listati (sia Basic che Assembler) rendono questo prodotto una vera e propria ghiottoneria non solo per chi è appassionato di Totocalcio ma anche per chi vuole impadronirsi di particolari tecniche di programmazione.

Abbinato alla pubblicazione è presente il nastro-cassetta contenente tutti i listati (non protetti) che compaiono sul fascicolo stesso; questi possono essere facilmente trasportati su disco con le consuete operazioni di Load e Save.

Prezzo: L. 8000 (cassetta compresa)

SPECIALE COMMODORE LINGUAGGIO MACCHINA

E' un fascicolo "speciale" dedicato a tutti coloro che si vogliono cimentare con il Linguaggio Macchina del C/64.

Contiene un corso completo sull'Assembler 6510, il microprocessore che anima il C/64, e il disassemblato commentato delle celeberrime "Routine di Toma" che aggiungono numerosi comandi al Basic V. 2.0.

Oltre ad essere un prodotto molto interessante per tutti coloro che programmano, si rivela addirittura un aiuto insostituibile per chi, esperto o principiante, voglia saperne di più sull'affascinante mondo del Linguaggio Macchina del piccolo Commodore.

Sono presenti numerosissimi programmi dimostrativi.

Assieme al fascicolo è possibile ottenere (a richiesta) il dischetto contenente tutti i programmi che compaiono sullo stesso fascicolo, in modo da evitarne la digitazione. Si ricorda che è assolutamente impossibile fornire il software su nastro.

Prezzo del fascicolo: L. 5000
Dischetto L. 12000

SHERLOCK HOLMES CLUB

Le avventure del più famoso investigatore di tutti i tempi.

Un fantastico dischetto per C/64 contenente diversi casi che impegneranno il nostro eroe.

Indizi, interrogatori, prove e, naturalmente... deduzione, nella tradizione di Sir Arthur Conan Doyle, ideatore di Sherlock Holmes.

Nel dischetto troverete due adventure animate: il caso Lancaster ed il caso Allan Fox.

Oltre alle adventure vi sono anche due videogame d'abilità: Grand Prix e Gara di pistola.

Prezzo: L. 12000 (solo su disco)

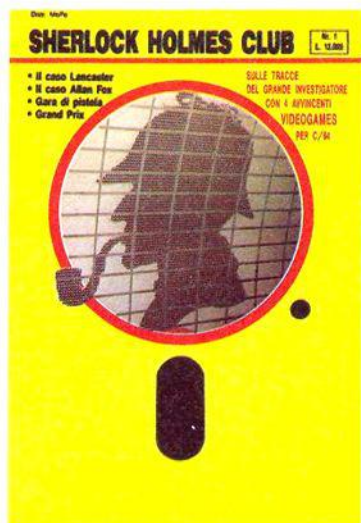
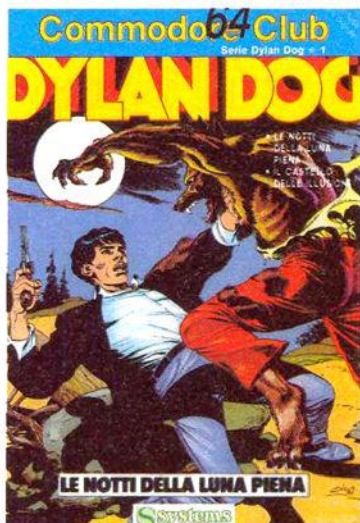
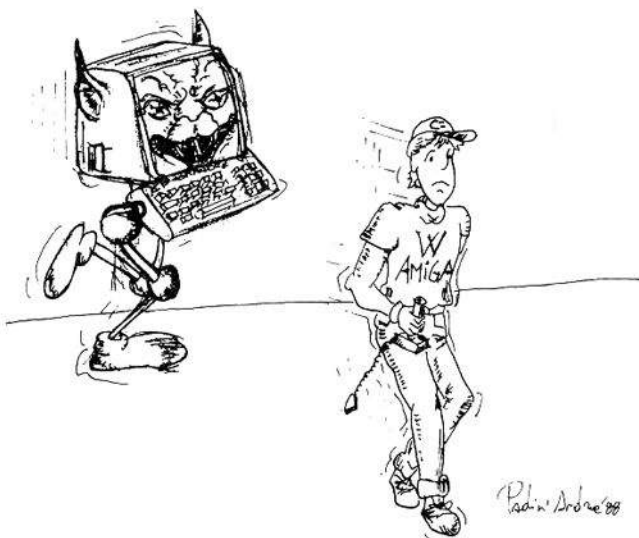
DYLAN DOG N. 1

Un videogame "da incubo" che vi trasporterà nella dimensione del mistero e dell'orrore.

Nei panni di Dylan Dog, il popolare eroe dei fumetti di Sergio Bonelli Editore, dovrete risolvere un intricato mistero condito con gli ingredienti più inquietanti.

Streghe e licantropi vi aspettano nel profondo della Foresta Nera per mettere alla prova il vostro coraggio e la vostra abilità.

Sul medesimo dischetto è presente anche un videogame che, nella consueta tradizione, vi vedrà armati di pistola contro ogni genere di nemico, umano e non.

Punto di riferimento per il "Software made in Italy", Dylan Dog vi aspetta in edicola ogni due mesi con una nuova avventura.

Prezzo: L. 15000 (solo su disco)

ATTENTI ALLE CATTIVE AMIGHIZIE.....

IL COMMODORE 64 EMULA L'AMIGA!

La gestione insolita del Raster consente di visualizzare ben 136 colori sullo schermo del vecchio C/64

di **Christoph Moar**

Chi di noi sessantaquattristi non si è mai soffermato davanti alla vetrina di un negozio di elettronica, per ammirare le stupende grafiche a 4096 colori del potente Amiga?

E chi, tornato a casa, si è rassegnato ai "soli" 16 colori del C/64?

Ma riflettiamo sulla tecnica con cui Amiga riesce a visualizzare 4096 colori in contemporanea:

Il computer provvede, semplicemente, a dimezzare la frequenza con cui le immagini vengono "spedite" al monitor, per creare due schermate, le cui linee vengono visualizzate alternativamente: una riga dalla schermata 1, una dalla schermata 2, quindi un'altra dalla schermata 1 e così via. Questa tecnica è denominata "Interlacing" (to interlace = interlacciare).

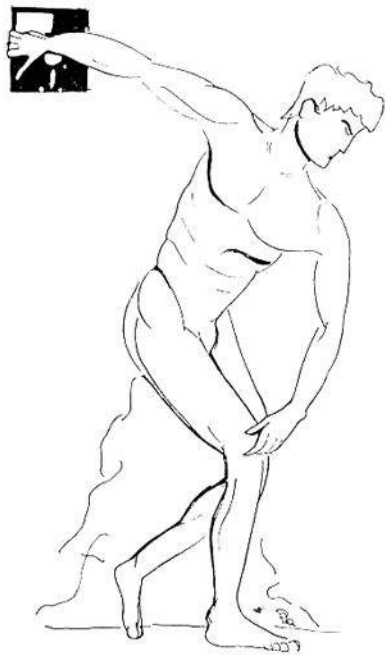
Se, dunque, l'Amiga ne è capace, perchè non dovrebbe poterlo fare anche il C/64? Ed è proprio questo compito che svolge il programma proposto: crea due false memorie colore, le cui informazioni vengono visualizzate alternativamente ad al-

SCHEDA TECNICA

Software didattico per applicazioni grafihe

Hardware richiesto: C/64; non adattabile ad altri computer Commodore

Ideale l'uso di un monitor a colori

Anche i programmi pubblicati in queste pagine sono contenuti nel disco "Directory" di questo mese.

tissima velocità. Risultato: i colori della mappa 1 si miscelano con quelli della mappa 2, grazie anche al fenomeno della persistenza dell'immagine propria dell'occhio umano.

"Ma in questo modo"- penserà il furbo lettore -"si dovrebbero visualizzare 16*16 = 256 colori!"

Il calcolo, purtroppo, è errato. Le combinazioni con i 16 colori, infatti, sono doppie, ed è ovvio: giallo e verde danno lo stesso colore che verde e giallo! Il furbo lettore (elementare, Watson) penserà dunque che i colori siano 256/2= 128. Ancora errato! Facendo così, infatti, si dividerebbero anche i colori originali (quelli che si otterrebbero mescolando lo stesso colore, come giallo e giallo, per intenderci)!

Ma, insomma, quant'è grande 'sta mappa colore? Ma è formata 136 colori, diamine! E chi non ci crede li potrà contare con il programma 2.

PER I PIU' ESPERTI

La tecnica usata nel programma (vedi disassemblato) è tanto semplice quanto geniale (modesti come noi non c'è nessuno): dopo avere bloccato qualsiasi tipo di interrupt, tranne quello di raster (vedi C.C.C. n. 47), il programma mette la riga n.255 dello schermo come "scatto" di interrupt (righe 260-450).

In pratica, ogniqualvolta il pennello elettronico del video raggiungerà la riga 255 ($FF), un "campanellino" suonerà nel C/64 ed a quel punto entra in azione la nuova routine di interrupt (480 - 870). Il campanellino viene fermato (490 - 500) e la routine decide se deve visualizzare la mappa colore 1 oppure la 2 (il counter viene controllato).

Nel caso che il programma debba mandare in video la finta mappa colore 1, la prima serie di colori viene copiata nella vera mappa colori (550- 650), il counter viene quindi messo ad uno, per assicurarsi che al prossimo interrupt venga visualizzata la seconda falsa memoria colore. La routine salta quindi a $EA31, cioè alla vera routine di interrupt.

Lo stesso vale per la seconda falsa memoria colore (720 - 870). Il programma potrebbe quindi già funzionare in questo modo, ma per cancellare uno schermo sarebbe necessario cancellare anche le due false memorie colore (allocate da $C100 a $C4E8 e da $C500 a $C8E8). Questo, in Basic, sarebbe possibile farlo soltanto tramite lentissime Poke. E' stata quindi implementata una routine che cancella le due mappe colori con i colori desiderati.

Osservando il disassemblato, è facilissimo capirne il funzionamento. Da menzionare soltanto la routine del S.O. denominata COMBYT, che controlla la presenza di una virgola (pena SYNTAX ERROR) e mette il numero successivo (da 0 a 255) in X. Cancellate le due mappe colori, non resta che uscire dalla subroutine.

MODALITA' D'USO

Anche per quei lettori che non abbiano compreso quanto detto, l'utilizzo della routine è semplicissimo: dopo aver inizializzato il tutto (SYS 49152), basterà cancellare le memorie colore con i colori di proprio gradimento (SYS 49270, colore1, colore2).

Qualsiasi cosa scriviate ora, verrà visualizzata in un colore ottenuto mescolando i due colori indicati. E' inoltre possibile pokare i colori per ogni casella video direttamente nelle due aree di memoria colore (allocate, rispettivamente, da 49408 a 50407 e da 50432 a 51431). Risulta quindi possibile realizzare, in Basic, effetti cromatici incredibili.

Sarà infatti sufficiente conoscere la linea e la colonna in cui mischiare i due colori ed usare i seguenti comandi:

POKE 49408 +l*40 +c, colore 1

...e...

POKE 50432 +l*40 +c, colore 2

IL DISCHETTO

...per ottenere i due colori sovrapposti. La variabile "L" indica il numero di linea mentre "C" la colonna desiderata.

IN PRATICA

Dopo aver digitato il caricatore Basic della routine (appena 162 byte) salvatelo accuratamente e date il RUN.

Apparirà una piccola spiegazione. Se volete cominciare subito con i vostri esperimenti... multicolori, impartite SYS 49152 e tutte le Poke necessarie. Se, invece, gradite sbalordirvi con le capacità della routine proposta, digitate anche il listato n. 2 (Demo) e date il RUN. Se ancora non crederete alla loro reale disponibilità, potrete contare, uno per uno, i 136 colori visibili, pur se con l'inevitabile sfarfallamento dovuto alla "lentezza" del microprocessore: in fin dei conti non è mica un 68000!

QUALCHE RIFLESSIONE

Con la routine proposta risulta impossibile (o quasi) multi-colorare i programmi già in commercio (vedi KOALA vari), ma non disperate: stiamo già pensando ad un programmino per ricolorare grafici in alta risoluzione con la tecnica descritta...

```
100 POKE53280,6:POKE53281,6:PRINT""
110 FORT=49152TO49313:READA:POKET,A:CS=CS+A:NEXT
120 READCH:IFCH<>CSTHENPRINT"ERRORE DI DIGITAZIONE"
130 PRINT"**********************"
140 PRINT"*                    *"
150 PRINT"* MAGIC COLOURS      *"
160 PRINT"*                    *"
170 PRINT"* BY CHRISTOPH MOAR *"
180 PRINT"*                    *"
190 PRINT"* 1988               *"
200 PRINT"*                    *"
210 PRINT"**********************"
220 PRINT:PRINT:PRINT"SYS 49152      > ATTIVA LA ROUTINE"
230 PRINT"SYS 49270,X,Y > CANCELLA LO SCHERMO"
240 PRINT:PRINT"POKE 49408 - 50407,X > MAPPA COLORE #1"
250 PRINT"POKE 50432 - 51431,Y > MAPPA COLORE #2"
260 PRINT:PRINT"X > COLORE 1"
270 PRINT"Y > COLORE 2"
280 DATA120,169,36,141,20,3,169,192,141,21,3,169,7,141,13,220,169,1,141,26
290 DATA208,169,255,141,18,208,173,17,208,41,127,141,17,208,88,96,173,25,208
300 DATA141,25,208,165,2,240,36,160,0,185,0,193,153,0,216,185,0,194,153,0,217
310 DATA185,0,195,153,0,218,185,0,196,153,0,219,136,208,229,169,0,133,2,76
320 DATA49,234,160,0,185,0,197,153,0,216,185,0,198,153,0,217,185,0,199,153
330 DATA0,218,185,0,200,153,0,219,136,208,229,169,1,133,2,76,49,234,32,0,226
340 DATA134,250,32,0,226,138,160,0,153,0,197,153,0,198,153,0,199,153,0,200
350 DATA136,208,241,165,250,153,0,193,153,0,194,153,0,195,153,0,196,136,208
360 DATA241,96,19724
370 END
```

```
10 SYS49152:SYS49270,1,1:POKE53280,0:POKE53281,0:PRINT"";:PO=5*40+12:P=207
20 FORZ=0TO15:READX:FORY=0TO15:POKE49408+Z*40+Y+PO,X
30 POKE50432+Z+Y*40+PO,X:POKE1024+Z*40+Y+PO,P:NEXT:NEXT
40 PRINTSPC(6)"                                     "
50 PRINTSPC(6)"  Y  HRISTOPH  OAR 1988  "
60 PRINTSPC(6)"                                     "
70 PRINTSPC(6)" 136 MAGICI COLORI...C-64 ! "
80 FORT=6TO33:POKE49408+T,7:POKE50432+T,7:NEXT
90 FORT=6TO33:POKE49448+T,7:POKE50472+T,13:NEXT
100 FORT=6TO33:POKE49488+T,13:POKE50512+T,13:NEXT
110 FORT=6TO33:POKE49528+T,13:POKE50552+T,3:NEXT
120 DATA1,15,12,11,0,7,10,2,8,9,13,3,5,14,4,6
130 POKE198,0:WAIT198,1:SYS49270,1,1:PRINTCHR$(147)
```

```
00120  ;136 colori sul
00130  ;magico c=64
00150  ;by christoph moar
00180  ;
00190          *=$c000     ;beginn of assembly
00200          .obj  *
00210  ;
00220 counter  = $02
00230 color1   = $fa
00250  ;
00260 setint   sei         ;dirotta
```

```
00270            lda #<intro ;vettore
00280            sta $0314   ;di
00290            lda #>intro ;interrupt
00300            sta $0315   ;ad 'intro'
00310  ;
00320            lda #$7f    ;blocca gli interrupt
00330            sta $dc0d   ;del cia 1
00340  ;
00350            lda #$01    ;libera interrupt
00360            sta $d01a   ;di raster
00370  ;
00380            lda #$ff    ;mette riga $ff
00390            sta $d012   ;come 'scatto' interrupt
00400            lda $d011   ;cancella bit 7
00410            and #$7f    ;che e' il bit 8 fittizio
00420            sta $d011   ;del registro di raster
00430  ;
00440            cli         ;fine routine
00450            rts         ;di wedge
00470  ;*** routine di interlacing ***
00490 intro      lda $d019   ;legge int. latch register
00500            sta $d019   ;e lo cancella
00510  ;
00520            lda counter ;legge contatore
00530            beq latch2  ;se contatore = 0 allora ==> latch2
00540  ;
00550 latch1     ldy #$00    ;azzera contatore
00560 loop1      lda $c100,y ;prende colore dalla mappa colore #1
00570            sta $d800,y ;e lo mette nella vera mappa
00580            lda $c200,y ;come sopra
00590            sta $d900,y ;
00600            lda $c300,y ;
00610            sta $da00,y ;
00620            lda $c400,y ;i bytes messi in piu'
00630            sta $db00,y ;non hanno significato
00640            dey         ;decrementa contatore
00650            bne loop1   ;se y <> 0 ==> loop1
00660  ;
00670            lda #$00    ;cancella
00680            sta counter ;contatore
00690  ;
00700            jmp $ea31   ;salta a routine di interrupt
00710  ;
00720 latch2     ldy #$00    ;come sopra
00730 loop2      lda $c500,y ;
00740            sta $d800,y ;
00750            lda $c600,y ;
00760            sta $d900,y ;
00770            lda $c700,y ;
00780            sta $da00,y ;
```

```
00790              lda $c800,y ;
00800              sta $db00,y ;
00810              dey         ;
00820              bne loop2   ;
00830   ;
00840              lda #$01    ;mette contatore
00850              sta counter ;uguale ad 1
00860   ;
00870              jmp $ea31   ;salta a routine di interrupt
00890   ;*** new clear screen ***
00910 clear        jsr $e200   ;combyt: check comma and get byte
00920              stx color1  ;byte ==> color1
00930              jsr $e200   ;come sopra
00940              txa         ;byte ==> accu
00950              ldy #$00    ;cancella contatore
00960 lp1          sta $c500,y ;mette il colore
00970              sta $c600,y ;nella mappa colore #2
00980              sta $c700,y ;
00990              sta $c800,y ;
01000              dey         ;decrementa contatore
01010              bne lp1     ;se y <> 0 ==> lp1
01020   ;
01030              lda color1  ;riprende primo colore
01040 lp2          sta $c100,y ;mette il colore
01050              sta $c200,y ;nella mappa colore #1
01060              sta $c300,y ;
01070              sta $c400,y ;
01080              dey         ;decrementa contatore
01090              bne lp2     ;se y <> 0 ==> lp2
01100   ;
01110   ;mappe colore cancellate !
01120   ;fine !
01130   ;
01140              rts
01150   ;
01160              .end

end of pass 1, 0 errors, * = $c0a2
end of pass 2, 0 errors, * = $c0a2


--- label file ---

counter = $0002      color1 = $00fa

setint = $c000       intro = $c024
latch1 = $c02e       loop1 = $c030
latch2 = $c052       loop2 = $c054
clear  = $c076       lp1   = $c081
lp2    = $c092
```

TRE MONITOR IN UNO

Come sfruttare tutti gli ingressi video dei monitor 1081 e 1084

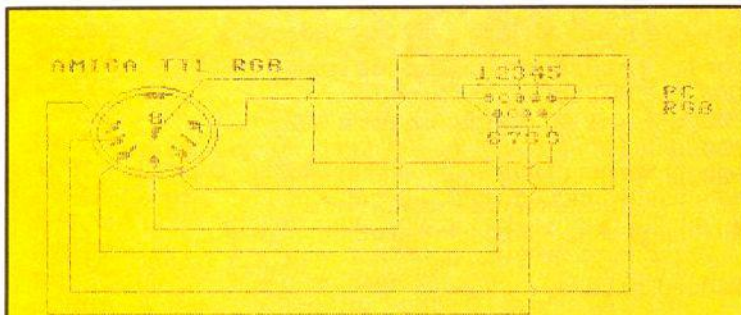
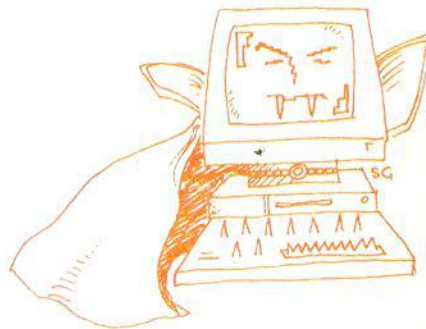
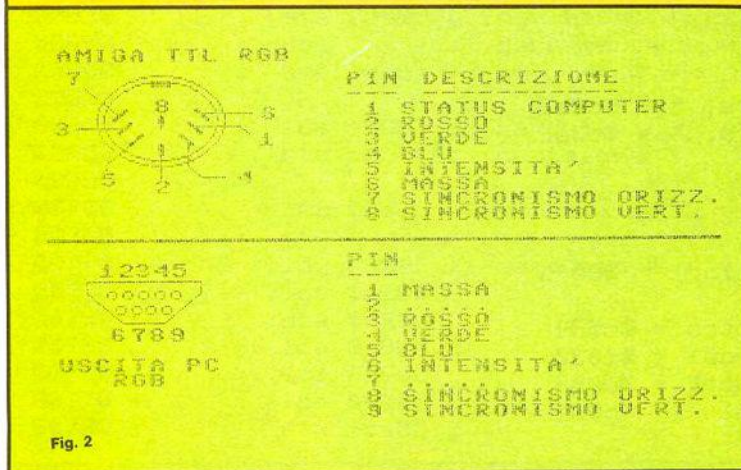
di **Antonio Pastorelli**

I monitor per Amiga 1081 e 1084 (eguale al 1081, ma dotato di schermo antiriflesso), posseggono tre ingressi video separati. Il primo, con presa Scart, è anche quello più noto ai possessori dei modelli Amiga.

E' previsto poi un ingresso video + audio, per i piccoli computer Commodore (C/64, C/128 in 40 colonne, C/16, Plus/4 e Vic-20), utilizzabile connettendo il normale cavo video in dotazione ai monitor per C/64 ed affini.

I problemi nascono quando si vuole utilizzare uno dei due monitor con un C/128 in 80 colonne o con un computer Ms-Dos compatibile. Il connettore posto sul retro di un PC o di un C/128, infatti, si differenzia da quello presente sul monitor per Amiga sia per la forma che per il numero di pin (vedi figura 1).

E' necessario quindi individuare, sui due connettori, i segnali che interessano e collegarli di conseguenza.

Il significato dei segnali corrispondenti ai vari pin è descritto nella tabella in cui sono evidenziati quelli relativi ai tre colori principali (rosso, verde e blu), all'intensità, alla massa ed ai sincronismi verticale ed orizzontale.

AMIGA TTL RGB

12345

6789

PC RGB

Fig. 1

AMIGA TTL RGB

PIN	DESCRIZIONE
1	STATUS COMPUTER
2	ROSSO
3	VERDE
4	BLU
5	INTENSITA'
6	MASSA
7	SINCRONISMO ORIZZ.
8	SINCRONISMO VERT.

12345

6789

USCITA PC RGB

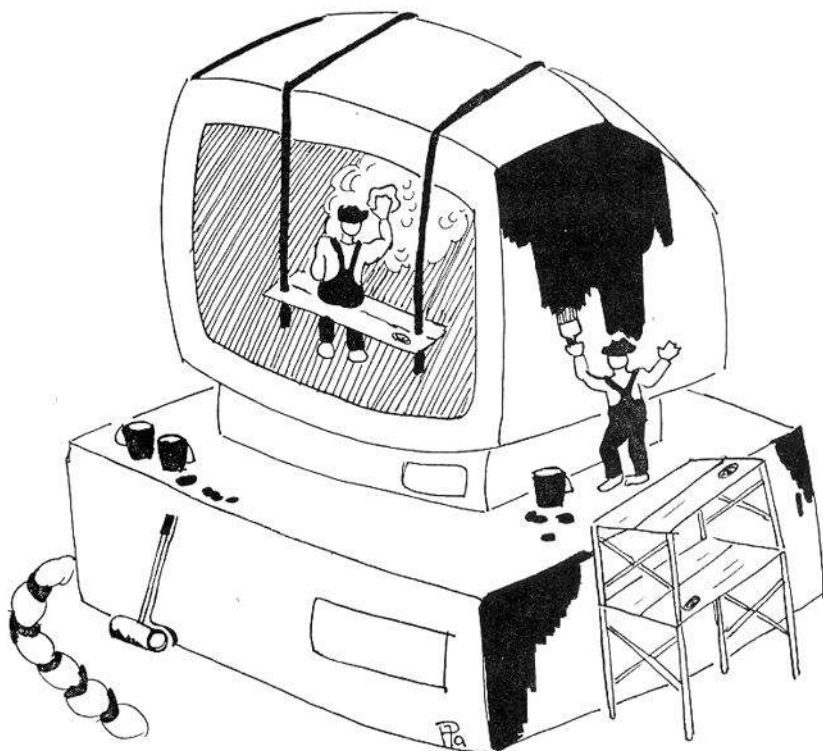
PIN	
1	MASSA
2	
3	ROSSO
4	VERDE
5	BLU
6	INTENSITA'
7	
8	SINCRONISMO ORIZZ.
9	SINCRONISMO VERT.

Fig. 2

La realizzazione del cavetto di connessione tra computer e monitor è molto semplice; basta prestare attenzione alla corrispondenza dei pin, tenendo presente che i connettori disegnati in figura 2 corrispondono alle prese (femmine) presenti sul monitor e sul computer, e non ai connettori (maschi) del cavetto.

Le necessarie saldature, da eseguire con cura particolare, completeranno il lavoro il cui costo, comprensivo di connettori e di 2 metri di cavo a 8 piste, non dovrebbe superare le 7.000 lire.

La realizzazione del progetto consente di collegare, ad uno stesso monitor, fino a tre computer, un Amiga, un Commodore 64 (o altri a 40 colonne), un C/128 (in modalità 80 colonne) oppure un PC.

Dal momento che i tre ingressi sono separati ed indipendenti, è possibile che i vari computer siano collegati contemporaneamente al monitor, evitando di rimuovere i cavi ogni qualvolta si utilizzi un computer diverso.

80 COLONNE SUL VOSTRO C/16 PLUS-4

Una semplice routine, che opera sullo schermo hi-res dei piccoli Commodore, permette di sviluppare insolite procedure

di **Doriano Rodighero**

Lo schermo di un computer, come è noto, consente di scrivere caratteri alfanumerici per una lunghezza di 40 colonne.

Operando in alta risoluzione, tuttavia, è possibile dimezzare (o quasi) la larghezza di ciascun carattere portando la matrice del carattere stesso a dimensioni più modeste.

Naturalmente tutto ciò è possibile grazie a due operazioni: la prima è il ricorso al linguaggio macchina, la seconda è l'utilizzo di un monitor, magari in bianco e nero. Un comune apparecchio TV, infatti, pur se perfettamente idoneo da un punto di vista strettamente tecnico, affaticherebbe la vista di chi decida di operare in 80 colonne.

Il programma di queste pagine, in effetti, rappresenta uno "studio" delle potenzialità offerte dal C/16 o dal Plus/4. Si tratta di un word processor dimostrativo che, privo di tutte le potenzialità offerte da prodotti analoghi, ha lo scopo preciso di dimostrare l'efficienza della routine l.m. proposta il cui disassemblato commentato è riportato in queste stesse pagine.

SCHEDA TECNICA

Software didattico per applicazioni semi-professionali

Hardware richiesto: C/16, Plus/4; non adattabile ad altri computer Commodore

Consigliato l'uso di un monitor

Consigliato agli esperti

Anche il programma pubblicato in queste pagine è contenuto nel disco "Directory" di questo mese.

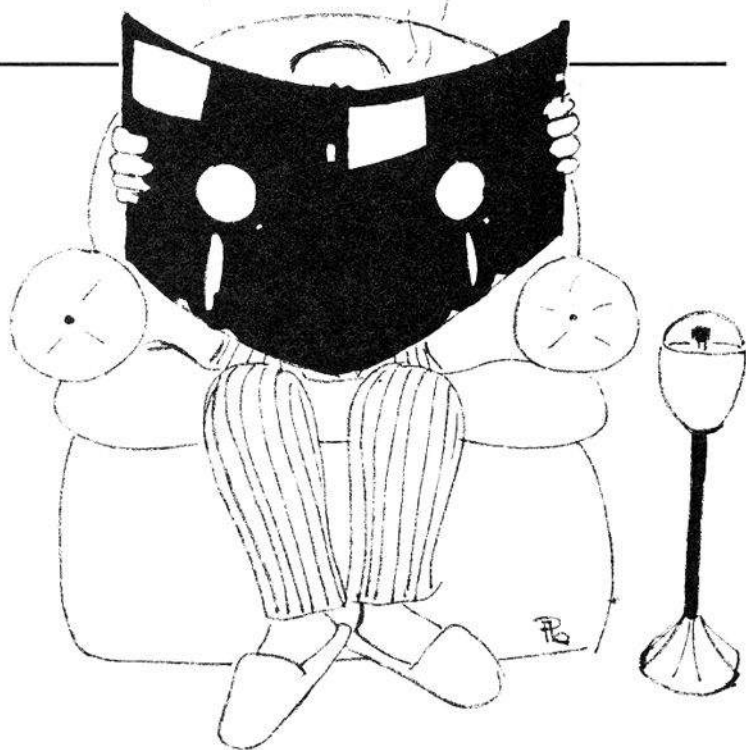
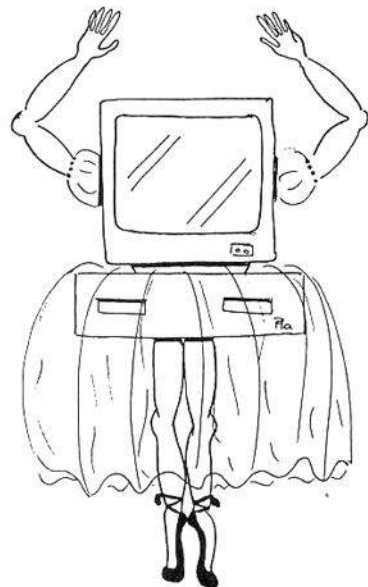
Chi volesse sfruttare la routine l.m. per riportare un testo di 80 colonne in hi-res potrà "estrarre" la routine stessa ed attivarla in propri programmi tenendo presente che la sintassi corretta è del tutto simile a quella Char (che inizia, nelle Rom del Plus/4, da $B9D4).

SYS in, co, ri, str, ind

...in cui:
In= locazione inizio routine (RILOCABILE!)
Co= colonna "virtuale" dello schermo (compresa tra 0 e 79).
Ri= riga video
Str= stringa da visualizzare
Ind= flag di reverse (1= reverse on; 0= reverse off)

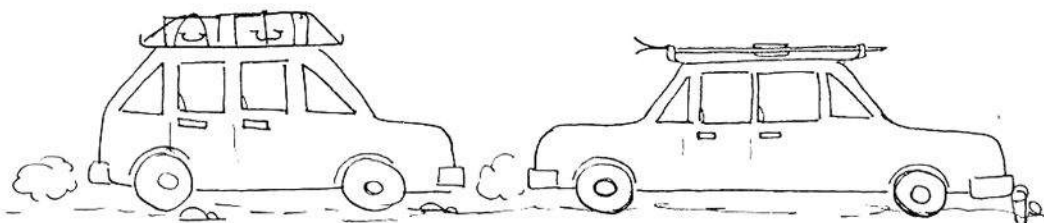
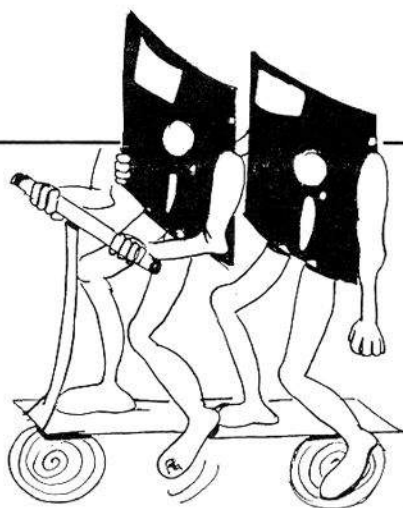
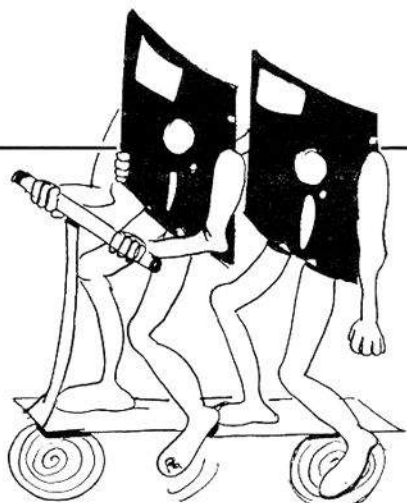
Per come è pubblicata, la routine è allocata in parte nella pagina video dal momento che, operando in hi-res, non occorre per le normali operazioni.

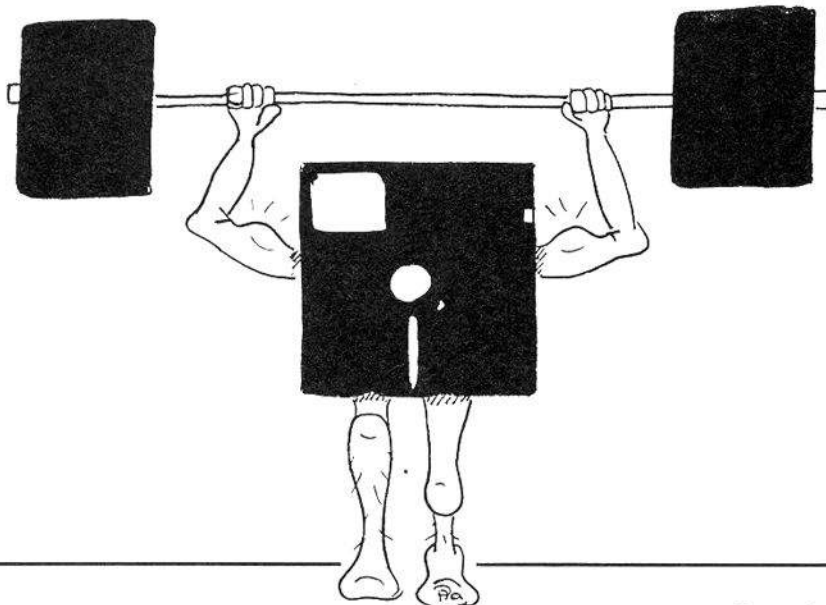
Il checksum dei Data non è riportato nel programma Basic. Si tenga presente, tuttavia, che la somma di tutti i valori contenuti nei Data è 31910 (vedi riga 134).

Benchè la routine sia destinata agli esperti, i principinati potranno utilmente digitarla, a patto di trascrivere il listato con la massima attenzione.

```
100 REM APPUNTI PER UN W/P PER C/16
101 REM E PLUS/4 SU 80 COLONNE
102 DATA 173,19,255,41,252,133,215,32
103 DATA 115,4,32,132,157,224,80,176
104 DATA 9,134,208,32,216,157,224,25
105 DATA 144,3,76,28,153,134,209,32
106 DATA 145,148,32,72,156,132,35,134
107 DATA 34,133,210,32,165,195,224,2
108 DATA 176,232,134,218,32,191,199,169
109 DATA 0,133,211,173,18,255,41,56
110 DATA 10,10,24,101,209,133,212,165
111 DATA 209,74,102,211,74,102,211,24
112 DATA 101,212,133,212,165,208,74,10
113 DATA 10,10,144,2,230,212,24,101
114 DATA 211,133,211,144,2,230,212,160
115 DATA 0,132,217,32,176,4,16,16
116 DATA 41,127,201,127,208,2,169,94
117 DATA 201,32,144,6,9,64,208,14
118 DATA 201,32,144,123,201,96,144,4
119 DATA 41,223,208,2,41,63,162,0
120 DATA 134,214,10,38,214,10,38,214
121 DATA 10,38,214,133,213,165,214,5
122 DATA 215,133,214,44,208,195,160,7
```

```
123 DATA 162,3,169,0,133,216,177,213
124 DATA 74,74,102,216,202,16,249,165
125 DATA 208,41,1,240,39,165,216,74
126 DATA 74,74,74,133,216,177,211,166
127 DATA 218,240,3,9,15,44,41,240
128 DATA 69,216,145,211,136,16,209,169
129 DATA 8,24,101,211,133,211,144,22
130 DATA 230,212,208,18,177,211,166,218
131 DATA 240,3,9,240,44,41,15,69
132 DATA 216,145,211,136,16,178,230,208
133 DATA 164,217,200,196,210,208,165,96
134 REM ******* CECKSUM = 31910 *******
135 PRINT"[illegible]"CHR$(27)"T"
136 IN=3072
137 FORX=INDTOIND+255:READA:POKEX,A:NEXT
138 REM PRINT"[illegible]DELETE-620":PRINT"[illegible]RUN"
139 REM POKE1319,19:POKE1320,13:POKE1321,13:POKE239,3
140 :
141 REM - INIZIO PROGRAMMA
142 :
143 DIM P$(24)
144 GRAPHIC1,1:PRINTCHR$(14)
145 FORY=1TO80:SP$=SP$+" ":NEXT
146 FORX=1TO24:P$(X)=SP$:NEXT
147 :
148 REM - SCRITT. TESTO
149 :
150 X=0:Y=0:C=1
151 DO
152 SYSIN,X,Y," ",1
153 GETKEYP$
154 A=(ASC(P$)AND127):IFA<32THENGOSUB162:GOTO158
155 MID$(P$(C),X+1)=P$
156 SYSIN,X,Y,P$
157 X=X+1:IFX=80THENX=0:IFC<24THENY=Y+1:C=C+1
158 LOOP
159 :
160 REM - DEL & INST/DEL
161 :
162 IFP$<>CHR$(20)THEN167
163 X=X-1:IFX<0THENX=0:GOTO166
164 A$=RIGHT$(P$(C),79-X)+" ":MID$(P$(C),X+1)=A$
165 SYSIN,0,Y,P$(C)
166 RETURN
167 IFP$<>CHR$(148)THEN175
168 A=INSTR(P$(C),"←",X+1):IFA=0ORA=80THEN171
169 A$=" "+RIGHT$(P$(C),80-X):MID$(P$(C),X+1)=LEFT$(A$,80-X)
170 SYSIN,0,Y,P$(C)
171 RETURN
```

```
172 :
173 REM - RETURN
174 :
175 IFP$<>CHR$(13)THEN183
176 A$="←"+LEFT$(SP$,79-X):MID$(P$(C),X+1)=A$
177 SYSIN,0,Y,P$(C)
178 X=0:IFC<24THENY=Y+1:C=C+1
179 RETURN
180 :
181 REM - TASTI CURSORE
182 :
183 IFP$<>"□"ANDP$<>"▨"ANDP$<>"▮"ANDP$<>"▮"THEN193
184 SYSIN,X,Y,MID$(P$(C),X+1,1)
185 IFP$="▮"THENX=X+1:IFX=80THENX=0:IFC<24THENY=Y+1:C=C+1
186 IFP$="▮"THENX=X-1:IFX<0THENX=0:IFC>1THENX=79:Y=Y-1:C=C-1
187 IFP$="▨"ANDC<24THENY=Y+1:C=C+1
188 IFP$="□"ANDC>1THENY=Y-1:C=C-1
189 RETURN
190 :
191 REM - HOME & CLR/HOME
192 :
193 IFP$<>"▤"THEN197
194 SYSIN,X,Y,MID$(P$(C),X+1,1)
195 Y=0:C=1:RETURN
196 :
197 IFP$<>"□"THEN203
198 SYSIN,X,Y,MID$(P$(C),X+1,1)
199 Y=24:C=24:RETURN
200 :
201 REM -
202 :
203 RETURN
```

```
90 rem        disassemblato commentato routine plot per il programma:
91 rem                 80 colonne sul video del c/16 e plus/4.
92 :
100 . 1c00  ad 13 ff lda $ff13 : carica valore attuale set di caratteri
101 . 1c03  29 fc     and #$fc  :
102 . 1c05  85 d7     sta $d7   : memorizza in $d7
103 . 1c07  20 73 04 jsr $0473 : preleva virgola dopo la sys
104 . 1c0a  20 84 9d jsr $9d84 : preleva nel registro x il numero di colonna
105 . 1c0d  e0 50     cpx #$50  : compara con #80
106 . 1c0f  b0 09     bcs $1c1a : se n. colonna >=#80 va a errore
107 . 1c11  86 d0     stx $d0   : altrimenti memorizza n. colonna in $d0
108 . 1c13  20 d8 9d jsr $9dd8 : preleva nel registro x il numero di riga
109 . 1c16  e0 19     cpx #$19  : compara con #25
110 . 1c18  90 03     bcc $1c1d : se n. riga <#25 continua
111 . 1c1a  4c 1c 99 jmp $991c : altrimenti vai a errore
112 . 1c1d  86 d1     stx $d1   : memorizza n. riga in $d1
113 . 1c1f  20 91 94 jsr $9491 : controlla presenza della virgola
114 . 1c22  20 48 9c jsr $9c48 : preleva la stringa e la lunghezza
115 . 1c25  84 23     sty $23   : byte hi puntatore alla stringa in $23
```

```
116 . 1c27  86 22     stx $22   : byte lo puntatore alla stringa in $22
117 . 1c29  85 d2     sta $d2   : lunghezza stringa in $d2
118 . 1c2b  20 a5 c3  jsr $c3a5 : preleva flag di reverse
119 . 1c2e  e0 02     cpx #$02  : compara con #2
120 . 1c30  b0 e8     bcs $1c1a : se flag >=#2 vai a errore
121 . 1c32  86 da     stx $da   : memorizza in $da flag di reverse
122 . 1c34  20 bf c7  jsr $c7bf : controlla se memoria hires e' allocata
123 . 1c37  a9 00     lda #$00  :
124 . 1c39  85 d3     sta $d3   :
125 . 1c3b  ad 12 ff  lda $ff12 : carica l'indirizzo della pagina grafica
126 . 1c3e  29 38     and #$38  :
127 . 1c40  0a        asl       :
128 . 1c41  0a        asl       : e lo somma
129 . 1c42  18        clc       :
130 . 1c43  65 d1     adc $d1   :   al
131 . 1c45  85 d4     sta $d4   :
132 . 1c47  a5 d1     lda $d1   : numero
133 . 1c49  4a        lsr       :
134 . 1c4a  66 d3     ror $d3   :   di
135 . 1c4c  4a        lsr       :
136 . 1c4d  66 d3     ror $d3   :  riga
137 . 1c4f  18        clc       :
138 . 1c50  65 d4     adc $d4   : moltiplicato per #320
139 . 1c52  85 d4     sta $d4   : pone il valore in $d3 e $d4
140 . 1c54  a5 d0     lda $d0   : carica numero colonna
141 . 1c56  4a        lsr       :
142 . 1c57  0a        asl       :
143 . 1c58  0a        asl       : lo moltiplica
144 . 1c59  0a        asl       :
145 . 1c5a  90 02     bcc $1c5e : per #4
146 . 1c5c  e6 d4     inc $d4   :
147 . 1c5e  18        clc       :
148 . 1c5f  65 d3     adc $d3   :
149 . 1c61  85 d3     sta $d3   : e lo somma
150 . 1c63  90 02     bcc $1c67 :
151 . 1c65  e6 d4     inc $d4   : in $d3, $d4
152 . 1c67  a0 00     ldy #$00  : azzera contacaratteri stringa
153 . 1c69  84 d9     sty $d9   :
154 . 1c6b  20 b0 04  jsr $04b0 : preleva un carattere della stringa
155 . 1c6e  10 10     bpl $1c80 :
156 . 1c70  29 7f     and #$7f  : routine
157 . 1c72  c9 7f     cmp #$7f  :
158 . 1c74  d0 02     bne $1c78 :   di
159 . 1c76  a9 5e     lda #$5e  :
160 . 1c78  c9 20     cmp #$20  : conversione
161 . 1c7a  90 06     bcc $1c82 :
162 . 1c7c  09 40     ora #$40  : carattere
163 . 1c7e  d0 0e     bne $1c8e :
164 . 1c80  c9 20     cmp #$20  :   da
```

```
165 . 1c82  90 7c     bcc $1cff  :
166 . 1c84  c9 60     cmp #$60   :  ascii
167 . 1c86  90 04     bcc $1c8c  :
168 . 1c88  29 df     and #$df   :    a
169 . 1c8a  d0 02     bne $1c8e  :
170 . 1c8c  29 3f     and #$3f   : codice schermo
171 . 1c8e  a2 00     ldx #$00   : routine
172 . 1c90  86 d6     stx $d6    :   di
173 . 1c92  0a        asl        : calcolo
174 . 1c93  26 d6     rol $d6    : della
175 . 1c95  0a        asl        : posizione
176 . 1c96  26 d6     rol $d6    :   del
177 . 1c98  0a        asl        : carattere
178 . 1c99  26 d6     rol $d6    : nella
179 . 1c9b  85 d5     sta $d5    : tabella
180 . 1c9d  a5 d6     lda $d6    : caratteri
181 . 1c9f  05 d7     ora $d7    : locata
182 . 1ca1  85 d6     sta $d6    : in $d000
183 . 1ca3  2c d0 c3  bit $c3d0  : maschera di salto
184 . 1ca6  a0 07     ldy #$07   : routine
185 . 1ca8  a2 03     ldx #$03   :   di
186 . 1caa  a9 00     lda #$00   : compattamento
187 . 1cac  85 d8     sta $d8    :   del
188 . 1cae  b1 d5     lda ($d5),y:
189 . 1cb0  4a        lsr        : carattere
190 . 1cb1  4a        lsr        :
191 . 1cb2  66 d8     ror $d8    : trovato
192 . 1cb4  ca        dex        :
193 . 1cb5  10 f9     bpl $1cb0  : il carattere compattato e' posto in $d8
194 . 1cb7  a5 d0     lda $d0    : carica n. colonna
195 . 1cb9  29 01     and #$01   : controlla se numero pari o dispari
196 . 1cbb  f0 27     beq $1ce4  : salta a stampa n. pari
197 . 1cbd  a5 d8     lda $d8    : carica carattere compattato
198 . 1cbf  4a        lsr        :
199 . 1cc0  4a        lsr        :
200 . 1cc1  4a        lsr        :
201 . 1cc2  4a        lsr        :  trasla tutto a destra
202 . 1cc3  85 d8     sta $d8    :
203 . 1cc5  b1 d3     lda ($d3),y:preleva attuale carattere
204 . 1cc7  a6 da     ldx $da    : controlla reverse
205 . 1cc9  f0 03     beq $1cce  :
206 . 1ccb  09 0f     ora #$0f   : esegui reverse
207 . 1ccd  2c 29 f0  bit $f029  :
208 . 1cd0  45 d8     eor $d8    : somma carattere trovato con compattato
209 . 1cd2  91 d3     sta ($d3),y:memorizza in attuale posizione
210 . 1cd4  88        dey        :
```

```
211 . 1cd5  10 d1      bpl $1ca8 :
212 . 1cd7  a9 08      lda #$08  : routine
213 . 1cd9  18         clc       :   di
214 . 1cda  65 d3      adc $d3   : incremento
215 . 1cdc  85 d3      sta $d3   :   della
216 . 1cde  90 16      bcc $1cf6 : posizione
217 . 1ce0  e6 d4      inc $d4   :    del
218 . 1ce2  d0 12      bne $1cf6 : carattere in hires
219 . 1ce4  b1 d3      lda ($d3),y:stampa del carattere pari
220 . 1ce6  a6 da      ldx $da   : controllo reverse
221 . 1ce8  f0 03      beq $1ced :
222 . 1cea  09 f0      ora #$f0  : esegui reverse
223 . 1cec  2c 29 0f bit $0f29 :
224 . 1cef  45 d8      eor $d8   :
225 . 1cf1  91 d3      sta ($d3),y:stampa carattere compattato
226 . 1cf3  88         dey       :
227 . 1cf4  10 b2      bpl $1ca8 :
228 . 1cf6  e6 d0      inc $d0   : incrementa numero colonna
229 . 1cf8  a4 d9      ldy $d9   :
230 . 1cfa  c8         iny       : incremento puntatore al carattere stringa
231 . 1cfb  c4 d2      cpy $d2   : controlla se fine stringa
232 . 1cfd  d0 a5      bne $1ca4 : se no allora continua
233 . 1cff  60         rts       :
234 :
235 ---------------------------------------------------------------------------
236 :
237 : $d0 n. colonna
238 : $d1 n. riga
239 : $d2 lunghezza stringa
240 : $d3;$d4 puntatore alla pagina grafica
241 : $d5;$d6 puntatore al carattere
242 : $d7 inizio tabella caratteri
243 : $d8 carattere compattato
244 : $d9 contacaratteri stringa
245 : $da flag per reverse
```

COMMODORE CAMBIA SEDE

L'area in cui erano prima sistemati gli uffici della Commodore Italiana, era troppo piccola per consentire lo svolgimento delle numerose attività che, di solito, sono indispensabili considerando le dimensioni di una Società internazionale.

La nuova sede (Viale Fulvio Testi 280, Milano) è grande quasi il doppio di quella precedente e favorirà l'espansione della prestigiosa azienda. Saranno quindi disponibili ben presto i nuovi posti di lavoro pianificati con la strategia impostata di recente.

L'ubicazione della nuova sede non è casuale: nei suoi dintorni (Ex-area Pirelli Bicocca), infatti, sono presenti gli uffici di aziende altrettanto note nel campo dell'informatica. Ciò dimostra, ancora una volta, la decisione del nuovo manager, l'ingegner Simonelli, di esser presente nel mercato del computer professionale a stretto contatto con la cosiddetta "concorrenza"; la possibilità di primeggiare anche in Italia, come già è avvenuto in Germania, diventa, ogni giorno che passa, una concreta realtà.

SMAU, CONFERMATO IL 2500

E' confermata la presenza allo SMAU, già anticipata nel numero scorso, del modello A-2500 che rappresenta, offrendo l'ambiente UNIX, l'alternativa Commodore in questo campo super-professionale.

Oltre a questo "mostro" saranno presentate varie schede, sempre per Amiga 2000, dotate di microprocessore 68020 (14.3 Mhz.) e co-processore matematico in virgola mobile 68881 (14 oppure 20 / 25 Mhz.). Ancora per Amiga (500 - 1000- 2000, purchè dotati di 1 mega Ram) potremo ammirare un monitor monocromatico (quattro livelli di grigio) ad alta risoluzione, da 15 pollici; vi sarà un Genlock semi-professionale (interno), un nuovo controller per hard disk SCSI (per A-2000), una scheda di espansione di memoria da 2 a 8 Mbyte. Finalmente dovrebbe esser disponibile la scheda per trasformare l'A-2000 in un computer Ms-Dos At-compatibile.

Tutte queste succose notizie, insomma, non fanno che confermare l'intenzione della Commodore, a livello internazionale, di sostenere con tutte le forze il favoloso computer Amiga che, ormai, rappresenta l'uncica alternativa valida per entrare in possesso, con una cifra modesta, di un sistema assolutamente unico per le potenzialità che offre.

CAMPER

Anche il camper Commodore, di cui abbiamo già parlato nel numero scorso, sarà presente allo Smau. Ciò che più importa, tuttavia, è che la notevole sofisticazione degli impianti ivi montati hanno suscitato l'interesse di molti Network televisivi che operano, in Italia, a livello nazionale. Presto vedremo, quindi, intere trasmissioni, ovviamente in diretta, supportate dall'ausilio "informatico" del complicato camper, appositamente attrezzato in tal senso.

COMMODORE CAMPIONE

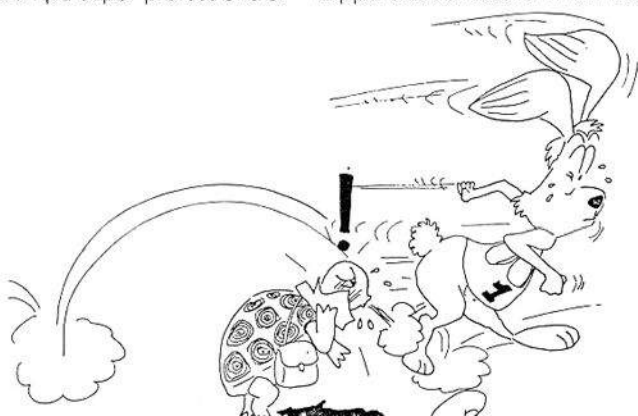
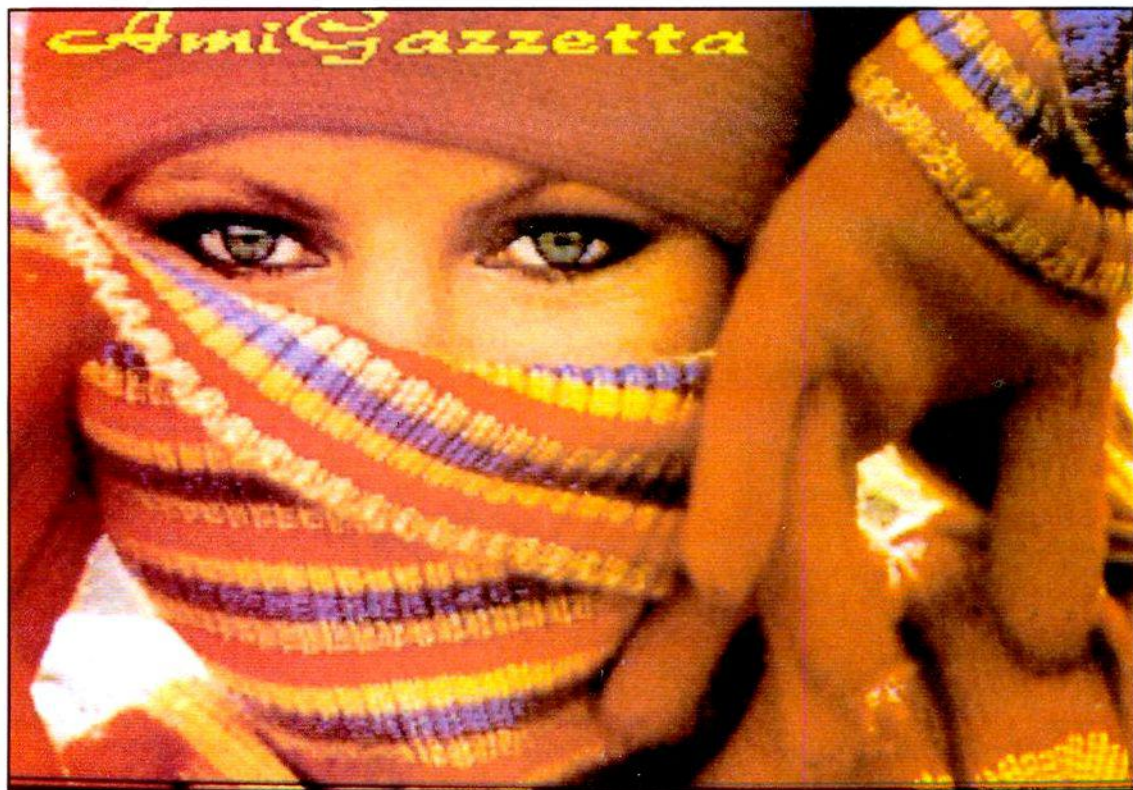
Ancora buone notizie dal fronte delle sponsorizzazioni sportive della Commodore. Il centauro Massimo Bonfante, che i nostri lettori conoscono da tempo, si è piazzato al quinto posto assoluto nella gara di Rijeka (Jugoslavia) disputata il 3 luglio scorso, e addirittura secondo nella classe 125, cilindrata della sua stupenda Honda. Grande interesse ha suscitato il furgone di "appoggio" di Massimo: tutti i gadget Commodore (cappellini, magliette, adesivi) sono andati a ruba nel giro di pochissime ore, segno questo che anche all'Est il nome Commodore è conosciuto come da noi.

UN AIUTO AI PIU' DEBOLI

Una recente notizia, pubblicata da numerosi periodici, ha commosso gli italiani: un bambino, portatore di handicap, ha ricevuto un notevole aiuto da un C/64 che, opportunamente programmato, ha potenziato la modesta capacità di comunicare del piccolo.

In seguito al successo dell'operazione, la Commodore ha deciso di offrirgli un intero sistema basato su Amiga ed ha affidato il compito di realizzare software specifico alla "Cloanto Italia" (Via G. B. Binson 24, Udine) che da anni si occupa esclusivamente di realizzare software per i portatori di handicap.

Il computer Amiga, infatti, si presta facilmente ad essere adoperato in tal senso, grazie alla facilità di gestione del mouse e, soprattutto, alla fedelissima sintesi vocale di cui dispone.

Sembra, inoltre, che un computer, opportunamente adattato, possa esser considerato come una "protesi" per i portatori di handicap e, come tale, rimborsabile, magari in parte, dal sistema sanitario nazionale.

AMIGAZZETTA TERZO

Nuove informazioni, fonti, sorgenti e programmi su dischetto interamente made in Italy

di **Luigi Callegari**

Il nuovo numero di AmiGazzetta, il terzo della serie, si presenta estremamente ricco e vario per i suoi contenuti. Ci riferiamo, in particolare, alle novità rispetto ai precedenti numeri. Il prezzo di vendita del dischetto da 3 pollici e 1/2 è rimasto invariato: solo L. 12000 (vedi riquadro per le modalità di ordinazione).

PROGRAMMI IN C

Altamente didattici, perchè commentati in italiano, sono presenti i programmi: HLINES, KC, SQ e USQ.

Il primo è un prezioso programma dimostrativo di grafica in alta risoluzione, ideale per vedere in pratica come si sfrutta la grafica di Amiga nel velocissimo linguaggio C.

KC è un listato semplicissimo, rivolto quindi ai super-principianti, idoneo per eliminare i commenti eventualmente presenti nei programmi sorgente. Gli ultimi file compongono un compattatore / decompattatore di file AmigaDOS. Si tratta di un programma estremamente sofisticato (suddiviso in due parti), che permette riduzioni anche del 60 % delle dimensioni di un file sul disco senza, ovviamente, perdere informazioni.

USQ serve, al contrario, a riportare, il file compattato, alle dimensioni originali. I due programmi sono trasportabili anche su altri sistemi (MP/M, CP/M, MS-DOS...). Di tutti questi programmi è presente il modulo eseguibile (CLI oppure Workbench) ed il testo sorgente originale.

PROGRAMMI BASIC

Intendiamo creare, con l'aiuto dei lettori di C.C.C, una libreria di FUNZIONI in AmigaBasic, sfruttando le caratteristiche di modularità di tale linguaggio. Per questa volta abbiamo inserito cinque nuove funzioni pronte per essere fuse (MERGE) nei vostri programmi: DisplayLED, Justify, Pinput, Qsort, Setcolor. Si tratta di funzioni piuttosto semplici, ma che sicuramente molti potranno trovare istruttive ed utili.

Tutte sono auto-documentate ed auto-dimostrative.

PROGRAMMI WORKBENCH

Accompagnati dalle loro brave icone, pronte per essere clickate, troviamo i programmi: FreeDraw, Gabryel's demo, Kaleido, Sector, Show, Speechtoy, Timer e Viruskill.

I primi tre sono semplici dimostrativi grafici (disegno a mano libera col mouse, generatori di figure colorate).

Sector visualizza istantaneamente la mappa di allocazione dei settori su dischetto dando alcune preziose informazioni (in italiano!).

Show è l'immancabile dimostratore di file IFF/HAM.

Speechtoy, invece, è un classico dimostrativo di sintesi vocale, con tutti i parametri controllabili da mouse.

Timer è un cronometro-workbench.

VirusKill è una nuova versione, spettacolare e completa, di eliminatore di virus per Amiga, commentata in italiano.

Tutti questi programmi sono accompagnati, come di consueto, da note sul loro uso corretto.

FONTI

Su "AmiGazzetta" n.3 vi sono inoltre, nella directory FONTS, ben sei nuove fonti di caratteri molto grandi e... scenografici. Ideali per creare paginate grafiche di titoli con Notepad oppure con programmi di Desk Top Publishing.

INFO

Nella directory "Istruzioni" vi sono 17 file clickabili di documentazione. Oltre alle informazioni d'uso di tutti i programmi (Basic, C, CLi e Workbench) presenti sul dischetto, sono stati inseriti degli indici (relativi ad articoli e programmi apparsi su Commodore Computer Club e su "AmiGazzetta") comodi da consultare, oltre ad alcuni articoli illustrativi leggibili con Notepad oppure, ovviamente, stampabili su carta.

Tra le altre cose viene spiegato come creare funzioni in Amigabasic (inviatecele dopo averle create!) in modo più dettagliato di quanto faccia il manuale originale. Molte curiosità, inoltre, sono riportate sui virus di Amiga.

Speriamo di avervi ingolosito abbastanza: "AmiGazzetta" assomiglia sempre di più ad una vera, seria e soprattutto originale rivista specializzata per Amiga, in italiano, su dischetto.

Ricordiamo anche che "AmiGazzetta" è aperta alla collaborazione di tutti i lettori: inviate fonti, programmi originali in qualunque linguaggio, icone ed informazioni; provvederemo a divulgare il materiale inviato.

PER AVERE GLI ARRETRATI

Per richiedere i prodotti Systems che mancano alla vostra collezione è sufficiente inviare al nostro indirizzo, in busta chiusa, un assegno bancario (di qualsiasi Banca, purchè non trasferibile) intestato a "Systems Editoriale - Milano".

La somma da indicare si può determinare riferendosi alle varie pagine informative, rintracciabili in questo stesso fascicolo, relative, ovviamente, ai SOLI prodotti Systems.

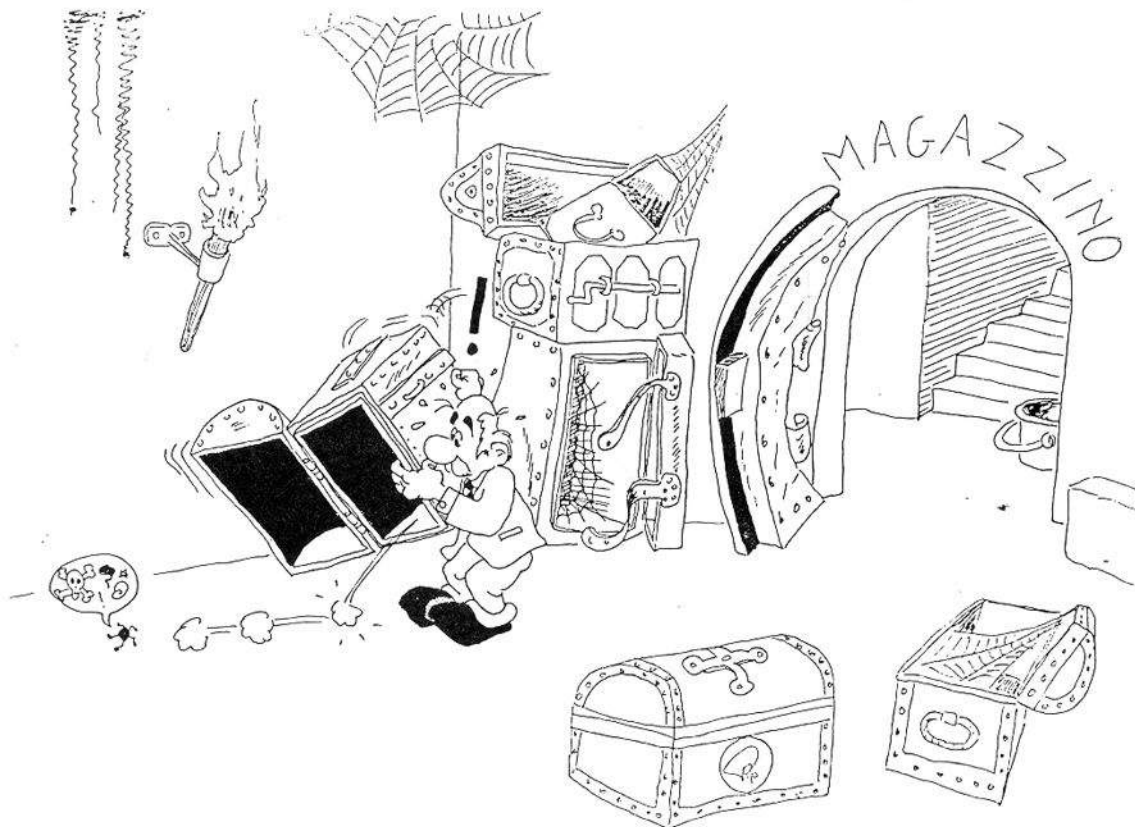
Alla cifra così calcolata si deve aggiungere la somma di L. 3000 per le spese di confezione e affrancatura (L. 6000 se si desidera la spedizione raccomandata); la cifra di L. 3000 si riferisce ad ogni spedizione effettuata, indipendentemente dal numero dei prodotti ordinati (dischi, libri, fascicoli arretrati e così via). Si devono aggiungere, insomma, L. 3000 sia per ordinare un solo disco che per ordinare l'intera produzione Systems.

Conviene, pertanto, richiedere più prodotti per volta, mettendosi in società, magari, con qualche amico.

Il lettore non dimentichi di inserire nella busta anche una lettera di accompagnamento in cui siano specificati i prodotti richiesti, oltre al proprio nominativo completo (cognome, nome, indirizzo, c.a.p. telefono).

Il nostro indirizzo è il solito:
Systems Editoriale
Servizio arretrati
Viale Famagosta 75
20142 Milano

Ricordiamo che non ci è assolutamente possibile inviare materiale contrassegno; si prega, quindi, di non insistere.

Se non disponete di un conto in banca, potete inviare la cifra richiesta utilizzando un normale modulo di conto corrente postale intestato a...
C/C N. 37952207
Systems Editoriale
Viale Famagosta 75
20142 Milano
...indicando sul retro del modulo (nello spazio riservato a: "causale del versamento") i prodotti richiesti ed il vostro nominativo completo.

IN CHE STATO SEI?

Spesso, utilizzando determinate Poke con il C/64, ci capita di non ricordare più a che punto siamo; allora...

di **Giancarlo Mariani**

Dopo svariati anni dall'uscita del C/64, anche i sassi dovrebbero sapere che questo computer (ma anche gli altri) ha bisogno, per il suo corretto funzionamento, di alcune locazioni di memoria, utilizzate come puntatori.

Nel caso del Commodore 64 queste locazioni di memoria si trovano nella cosiddetta pagina zero, ossia nei primi 256 byte della memoria Ram.

Proprio per questo motivo (memorizzazione in Ram) la loro modifica da parte del sistema operativo, o del programmatore, risulta molto facile.

In queste fatidiche particelle di memoria sono contenute, infatti, tutte le informazioni che necessitano al C/64 per il mantenimento in memoria di programmi Basic, variabili, e così via.

Ad esempio nella locazione 43 (decimale) è contenuto il byte basso dell'inizio della memoria e, quindi, del programma Basic; nella locazione 44 è contenuto il byte alto. Se vogliamo determinare l'indirizzo iniziale della memoria, non dovremo far altro che scrivere...

```
Print Peek (43) + Peek (44)*256
```

Allo stesso modo le locazioni 45 e 46 contengono la fine del programma Basic, 55 e 56 il limite della memoria, sempre per il Basic, e così via.

Per visualizzare tutte queste informazioni è sufficiente ripetere l'operazione sopracitata per ogni coppia di byte che interessa. A lungo andare, però, questo sistema diventa piuttosto noioso, e risulta certo più comodo averle a disposizione con un solo comando.

Detto... fatto. La semplice routine l.m. di queste pagine legge e visualizza tutte le locazioni utili per lo svilup-

Per chi inizia

Ricordiamo, ai nuovi lettori che ci leggono per la prima volta, che la nostra rivista propone su ogni numero un gruppo di routine in Linguaggio Macchina, per il Commodore 64, interamente rilocabili. Ciò significa che l'utente potrà realizzare una vera e propria enciclopedia in L.M. personalizzata in base alle proprie esigenze.

E' ovvio che le routine proposte, per funzionare adeguatamente, soggiacciono ad alcuni limiti, ben illustrati nei primi numeri di "Nuovo Sistema".

In questa sede ci limitiamo a ricordare che, per evitare malfunzionamenti, è necessario:

- **Fissare il Top di memoria Ram a 20000.**
- **Caricare, servendosi del programma "Caricatore" a suo tempo pubblicato (o analoghi), le routine Basic contenenti le istruzioni Data.**
- **Indicare la prima locazione di memoria in cui si desidera allocare la routine stessa.**
- **Attivare la routine secondo i suggerimenti indicati nei Demo a corredo.**

po di programmi Basic, in modo da rendere disponibili all'operatore le informazioni più importanti sullo stato del sistema; ecco perchè la routine porta il nome "System", non perchè è stata pubblicata su di una rivista della Systems Editoriale...

UN'AGGIUNTA

Non contenti delle informazioni che si potevano ottenere, abbiamo deciso di aggiungere un altro "pezzo", che controlla il collegamento, o meno, di tutte le varie periferiche collegate al computer.

Per utilizzare la routine non dovete far altro che caricarla in memoria da un indirizzo qualsiasi (dato che è rilocabile, come tutte le routine dell'enciclopedia) e quindi impartire la SYS relativa all'indirizzo di partenza: non necessitano, infatti, altri parametri.

Ammettendo di non aver commesso errori di digitazione, dopo la SYS compare una "maschera", che indica tutte le informazioni elencate qui di seguito:

A: Inizio programma Basic (Peek 43+Peek 44*256)
B: Fine programma Basic, Inizio variabili (Peek 45 + Peek 46*256)
C: Inizio Arrays (Peek 47 + Peek 48*256)
D: Fine Arrays (Peek 49 + Peek 50*256)
E: Inizio memoria per stringhe (Peek 51 + Peek 52*256)
F: Puntatore di utilità per stringhe (Peek 53 + Peek 54*256)
G: Limite memoria per programma Basic e variabili (Peek 55 + Peek 56*256)
H: Numero linea Basic appena trattata (Peek 59 + Peek 60*256)
I: Indirizzo per eventuale CONT (Peek 61 + Peek 62*256)
J: Eventuale numero di linea DATA corrente (Peek 63 + Peek 64*256)
4: Periferica 4 (stampante). *=accesa
5: Periferica 5 (stampante). *=accesa
8: Periferica 8 (drive). *=acceso
9: Periferica 9 (drive). *=acceso

In pratica, le lettere da A a J ed i numeri 4, 5, 8, 9, vengono stampati prima di ogni indirizzo visualizzato. L'utente deve solo ricordare che alla lettera A corrisponde l'inizio del programma Basic, alla lettera B la fine, e così via; oppure riferirsi alla tabella.

Di fianco ai numeri, che indicano il numero di periferica in esame, può comparire un asterisco o... niente. Se c'è un asterisco significa che la periferica in questione risulta accesa, altrimenti è considerata spenta (o non collegata).

Un esempio di videata potrebbe essere il seguente:

A:2049
B:4573
C:5600
....
....
J:10020
4:*
5:
8:*
9:

```
1 REM ** SYSTEM : VERSIONE BASIC
2 REM ** BY MARIANI G.
3 :
10 L=43
20 A=65
30 PRINT
40 PRINT CHR$(A);":";PEEK(L)+PEEK(L+1)*256
50 L=L+2
60 IF L=57 THEN L=L+2
70 A=A+1
80 IF A<75 THEN 40
90 P=4
110 OPEN 7,P,15
120 POKE 781,7:SYS 65481
130 SYS 65463:S=PEEK(780)
140 SYS 65484:CLOSE 7
142 S$=" "
144 IF S=0 THEN S$="*"
150 PRINT RIGHT$(STR$(P),1);":";S$
160 P=P+1:IF P=6 THEN P=P+2
170 IF P<10 THEN 110
180 END
```

A tutto disco.

Finalmente, viste le numerose richieste, d'ora in poi ogni pubblicazione ***Software Club*** *su cassetta sarà disponibile anche su dischetto da richiedersi, per corrispondenza, presso la redazione.*

Sono disponibili i seguenti titoli:

Software Club #11 *(C/64-128, C/16, Plus/4 e Vic 20)*
Software Club #12 *(C/64-128, C/16, Plus/4 e Vic 20)*
Software Club #13 *(C/64-128, C/16, Plus/4 e Vic 20)*
Software Club #14 *(C/64-128, C/16, Plus/4 e Vic 20)*

I Gialli Commodore *(C/64-128)*
I Gialli Commodore 2 *(C64/C128)*
Charlie Deus *(C/64-128)*
La voce III *(C/64-128)*
Commodore 64 Club *(C64)*

Il prezzo, per ognuna delle suddette pubblicazioni è di L. 12.000 più L. 3.000 per spese di spedizione.

Le richieste vanno indirizzate a:

Systems Editoriale
Viale Famagosta, 75
20142 MILANO
Tel. 02/8467348

Il pagamento può essere effettuato tramite assegno bancario o versamento sul c/c N. 37952207
Non è possibile inviare materiale in contrassegno nè contro invio di vaglia telegrafico.
Per ogni ordine, anche se per più dischetti, le spese rimangono fissate in L. 3.000

Sempre un passo avanti.

```
1000 PRINT CHR$(147);"ROUTINE SYSTEM"
1010 PRINT:PRINT"SYS XXXX"
1020 RETURN
1100 DATA 169,065,133,002,169,043,133,253,169,000,133,254,160,000,169,013,032
1110 DATA 210,255,192,014,208,002,200,200,165,002,032,210,255,169,058,032,210
1120 DATA 255,177,253,170,200,177,253,200,132,251,032,205,189,164,251,230,002
1130 DATA 169,013,032,210,255,165,002,201,075,208,213,169,004,133,251,165,251
1140 DATA 032,177,255,169,007,166,251,160,000,032,186,255,169,000,032,189,255
1150 DATA 032,192,255,169,000,032,147,255,032,183,255,133,002,032,132,255,169
1160 DATA 007,032,195,255,166,251,169,000,032,205,189,169,058,032,210,255,169
1170 DATA 032,166,002,208,002,169,042,032,210,255,169,013,032,210,255,230,251
1180 DATA 165,251,201,006,208,004,230,251,230,251,201,010,208,172,169,013,076
1190 DATA 210,255,-1,22361
```

In questo caso, tralasciando i vari indirizzi, significa che la stampante (4) ed il disco (8) sono accesi.

Il funzionamento della routine è molto semplice, dato che non fa altro che sfruttare alcune routine del S.O. per leggere, e visualizzare, i vari dati; l'unico lavoro "manuale" da fare è quello di tenere i vari contatori necessari.

Per ciò che riguarda la seconda parte della routine, relativa al controllo delle periferiche, viene ancora sfruttato il S.O, dapprima inviando un segnale di LISTEN alla periferica in esame, quindi aprendo un canale sulla stessa per controllarne lo stato (STATUS) al ritorno dalle varie routine.

Un'idea più precisa del funzionamento, comunque, potrete averla esaminando il disassemblato commentato e la versione in Basic della stessa routine.

Un appunto è da fare sul controllo della stampante: se possedete una stampante dotata di deviatore On / Off line, questo, se la stampante è accesa, dovrà essere su On, altrimenti la routine System andrà in Loop infinito aspettando, appunto, che si liberi la "linea" della stampante.

```
8000 lda #$41      ;Inizializza car. "A"
8002 sta $02       ;Lo salva
8004 lda #$2b      ;Inizializza la locaz.
8006 sta $fd       ;da cui partire a
8008 lda #$00      ;leggere (43).
800a sta $fe       ;
800c ldy #$00      ;
800e lda #$0d      ;Lascia una riga vuota
8010 jsr $ffd2     ;
8013 cpy #$0e      ;Controlla Y x salto
8015 bne $8019     ;Se y=14, y=y+2
8017 iny           ;
8018 iny           ;
8019 lda $02       ;Stampa il carattere
801b jsr $ffd2     ;
801e lda #$3a      ;Stampa i ":"
8020 jsr $ffd2     ;
8023 lda ($fd),y   ;Legge Dalla locazione
8025 tax           ;Trasferisce LO in X
8026 iny           ;
8027 lda ($fd),y   ;Legge dalla loc+1
8029 iny           ;(HI in A)
802a sty $fb       ;Salva contatore
802c jsr $bdcd     ;Stampa LO+HI*256
802f ldy $fb       ;Ripristina contatore
8031 inc $02       ;Incrementa carattere
8033 lda #$0d      ;Stampa un RETURN
8035 jsr $ffd2     ;
8038 lda $02       ;Controlla se
803a cmp #$4b      ;carattere="J"
803c bne $8013     ;Se no, continua
803e lda #$04      ;Altrimenti, perifer=4
8040 sta $fb       ;Salva periferica
8042 lda $fb       ;Invia un LISTEN
8044 jsr $ffb1     ;alla periferica.
8047 lda #$07      ;Apre il canale 7
8049 ldx $fb       ;sulla periferica
804b ldy #$00      ;
804d jsr $ffba     ;Setta i vari
8050 lda #$00      ;parametri.
8052 jsr $ffbd     ;
8055 jsr $ffc0     ;Apre il canale
8058 lda #$00      ;
805a jsr $ff93     ;Invia indirizzo sec.
805d jsr $ffb7     ;Legge lo status
8060 sta $02       ;Lo salva in $02
8062 jsr $ff84     ;Inizializza I/O
8065 lda #$07      ;Chiude canale 7
8067 jsr $ffc3     ;
806a ldx $fb       ;Stampa il numero
806c lda #$00      ;periferica
806e jsr $bdcd     ;
8071 lda #$3a      ;Stampa ":"
8073 jsr $ffd2     ;
8076 lda #$20      ;Car.in A uno spazio
8078 ldx $02       ;Controlla status
807a bne $807e     ;Se periferica accesa
807c lda #$2a      ;Mette in A un "*"
807e jsr $ffd2     ;Stampa A
8081 lda #$0d      ;Stampa un RETURN
8093 jsr $ffd2     ;
8086 inc $fb       ;Incrementa periferica
8088 lda $fb       ;
808a cmp #$06      ;Se e'=6, allora
808c bne $8092     ;
808e inc $fb       ;la incrementa di
8090 inc $fb       ;due volte
8092 cmp #$0a      ;Se e'<10 continua
8094 bne $8042     ;
8096 lda #$0d      ;Altrimenti stampa
8098 jmp $ffd2     ;un RETURN ed esci.
```

commodore
COMPUTER
CLUB
La rivista degli utenti di sistemi Commodore
games
TIME WARP

Come leggere le recensioni

Ogni mese, su queste pagine, verranno esaminati e testati i videogame più recenti per i computer Commodore 64 ed Amiga.

Ad ogni descrizione verranno associate una immagine, catturata tra le più belle schermate, ed una breve pagella.

Quest'ultima, pur se, inevitabilmente, frutto di impressioni personali di chi esamina il gioco stesso, ha lo scopo di assegnare una valutazione del livello del software,soprattutto tenendo conto di altri game analoghi disponibili sul mercato.

La pagella, comprende cinque voci:

IMPATTO: indica il livello di interesse suscitato dalla presentazione e dal tema del gioco.

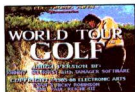
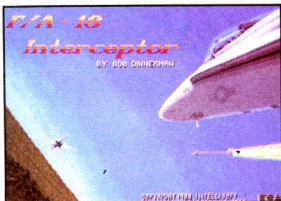
SCENARIO: riguarda l'accuratezza con cui è realizzata la grafica e l'efficacia dei disegni degli sprite.

SUONO: valuta gli effetti sonori presenti e le eventuali musiche di sottofondo.

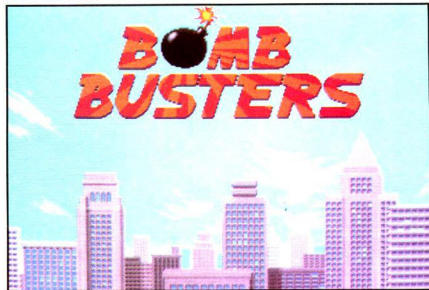
INTERESSE: si riferisce al livello di interesse che il gioco può suscitare in un giocatore abituato ai videogame e, implicitamente, alla sua probabilità di "permanenza" sui vostri monitor.

TOTALE: ha lo scopo di sintetizzare i precedenti valori con un voto unico.

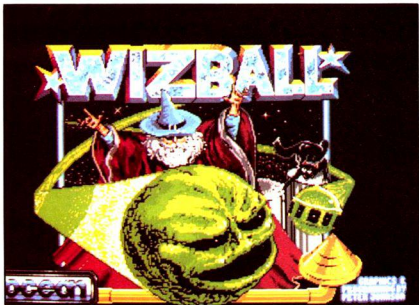
WORLD TOUR GOLF
Amiga - C/64

Una nuovissima versione di Golf, con grafica forse inferiore all'originale vecchio venduto dalla Commodore, ma più variegato e tecnicamente più apprezzabile dagli esperti. La grafica è di dimensioni piccole e l'animazione molto semplificata. La velocità di manovra è alta.

IMPATTO:	7
SCENARIO:	7
SUONO:	4
INTERESSE:	5
TOTALE:	**6**

F18 INTERCEPTOR
Amiga

E' forse la più bella simulazione di volo aereo mai vista su di un personal computer. La tecnica grafica segna nuovi limiti delle capacità di Amiga (fra l'altro il programma sfrutta l'espansione ad 1 mega, se presente). E' possibile allenarsi con un aereo pilota nelle evoluzioni e poi anche combattere.

IMPATTO:	9
SCENARIO:	9
SUONO:	9
INTERESSE:	9
TOTALE:	**9**

BOMB BUSTERS
Amiga

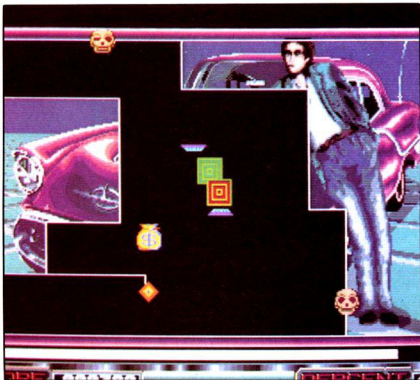
Null'altro che una versione Amiga del vetusto Bomb Jack per C/64. Si manovra un omino, per un numero incalcolato di schermi singoli (lo sfondo varia al completamento di una fase, ed è sempre molto bello graficamente), che deve disattivare, a suon di salti, delle bombe disseminate per lo schermo. L'idea è semplice ed altrettanto banale. La grafica, pur se valida, spesso non c'entra nulla con il gioco.

IMPATTO:	7
SCENARIO:	8
SUONO:	8
INTERESSE:	6
TOTALE:	**7**

WIZBALL
Amiga - C/64

Guidare una palla rimbalzante su di un pianeta alieno è un compito davvero ingrato, a dispetto della grafica e del realismo di questa produzione Ocean veramente entusiasmante. E' difficile iniziare a causa delle difficoltà di controllo dei rimbalzi e per il numeri di ostacoli animati e di insidie; altrettanto difficile è smettere.

IMPATTO:	*9*
SCENARIO:	*8*
SUONO:	*7*
INTERESSE:	*8*
TOTALE:	**8**

POWER STYX
Amiga

Un gioco veramente originale e divertente. Si manovra un quadratino che, delimintando figure chiuse, scopre disegni sottostanti. Quando si è scoperto il 75% di un disegno si procede con un altro. Ovviamente vi sono molti nemici che si aggirano per lo schermo e varie modalità per incrementare il punteggio. Comunque un gioco superconsigliato.

IMPATTO:	*8*
SCENARIO:	*9*
SUONO:	*8*
INTERESSE:	*8*
TOTALE:	**8**

GRAVE DIVING
Amiga - C/64

Un classico gioco tipo Rambo I, dove un militarino (alto 8 pixel o poco più!) deve correre e sparare a tutti. L'idea del gioco fa quasi pena, ma la grafica di sfondo è pregevole (scorrimento a tre strati, con rifinitura grafica eccellente); le figure in movimento sono, però, assolutamente ridicole, mentre gli effetti sonori sfiorano il patetico e, spesso, vi cadono dentro.

IMPATTO:	*6*
SCENARIO:	*6*
SUONO:	*4*
INTERESSE:	*6*
TOTALE:	**5**

SENTINEL
Amiga

Un classico gioco con prospettiva tridimensionale estremamente realistica. Col mouse ci si aggira in tempo quasi-reale per un mondo fatto a scalini, piante, discese ed insidie, molto statico.

Richiede sicuramente molto tempo per essere capito e goduto: è quindi consigliato solo ai videogiocatori più esasperati.

IMPATTO:	*9*
SCENARIO:	*8*
SUONO:	*5*
INTERESSE:	*6*
TOTALE:	**6**

FUTURE TANKS
Amiga

Un gioco concettualmente semplice, basato sulla grafica extra di Amiga. Si manovra un carro armato su di un pianeta invaso da mezzi terrestri alieni. Bisogna sparare a tutto e raccogliere oggetti utili. La scenografia è grande molte volte lo schermo e rifinita in maniera pregevole. L'idea non è originale, ma realizzata tecnicamente in maniera ottimale.

IMPATTO:	*8*
SCENARIO:	*8*
SUONO:	*8*
INTERESSE:	*6*
TOTALE:	**8**

BOOMERAID
Amiga

Siete un guardiano australiano che deve difendere i propri possedimenti da cattivoni a suon di boomerang, raccogliendo oggetti, eccetera. La grafica è di piccole dimensioni, ma molto rifinita tecnicamente. Quando si vuole lanciare il boomerang, sembra più di manovrare un CAD che un videogioco, a causa dell'elevato numero di parametri presenti.

IMPATTO:	*8*
SCENARIO:	*8*
SUONO:	*6*
INTERESSE:	*8*
TOTALE:	**8**

EXCALIBUR
Amiga

Non lasciatevi imbrogliare dal nome medievale, perchè si tratta dell'ennesima versione di uno scrolling-zapper tipo Uridium. In pratica si tratta della solita astronavazza che si aggira e sparacchia lungo una piattaforma infinita a scorrimento fluido verticale. La grafica ed i suoni sarebbero gradevoli, ma l'ideazione si riferisce a schemi ormai nauseanti. Consigliabile solo ai forti di stomaco.

IMPATTO:	*6*
SCENARIO:	*8*
SUONO:	*7*
INTERESSE:	*6*
TOTALE:	**7**

SCREAMING WINGS
Amiga

Uno scrolling-zapper a scorrimento verticale, ove invece di un'astronave (come in Uridium per C/64) si manovra un aereoplano che incrocia altri velivoli molto bellicosi, sorvolando boschi e ruscelli. La grafica è buona ma animata rozzamente. L'idea, inoltre, è vecchia.

IMPATTO:	*5*
SCENARIO:	*7*
SUONO:	*5*
INTERESSE:	*5*
TOTALE:	**5**

GUNSHOT
Amiga - C/64

Un gioco classicissimo. Si fa la guardia in una banca e bisogna sparare ai rapinatori che appaiono dalle tre porte senza colpire i clienti. Si richiede dunque rapidità e colpo d'occhio. Vi sono anche fasi intermedie di duello puro stile "mezzogiorno di fuoco". La grafica è ineccepibile, il caricamento addirittura turbo.

IMPATTO:	*7*
SCENARIO:	*7*
SUONO:	*6*
INTERESSE:	*7*
TOTALE:	**7**

BUGGY BOY
Amiga

Un videogioco classico, evoluzione del classico Penguin dell'Atari. Si manovra un bolide fuoristrada su percorsi da incubo tipo "Pole Position" ma con alberi sulla strada, bandiere da raccogliere, tunnel da attraversare, eccetera. Il punto di forza è sicuramente la grafica, decisamente all'altezza di una sala giochi.

IMPATTO:	*8*
SCENARIO:	*9*
SUONO:	*8*
INTERESSE:	*7*
TOTALE:	**8**

QUANTO COSTA IL TUO COMMODORE

LISTINO PREZZI HOME COMPUTER

Codice	Prodotto	Prezzo (IVA esclusa)
	Personal Computer	
A 500	Personal Computer 16/32 BIT CPU 512K RAM incorpora un floppy disk drive da 3" 1/2 da 880Kb e un mouse	**L. 950.000**
A 501	Espansione di memoria per Amiga 500 512Kbytes e orologio	**L. 212.500**
A 520/1	Modulatore per Amiga 500 permette di collegare Amiga 500 ad un qualsiasi televisore b/n e colori	**L. 40.000**
A SCART	Cavo collegamento Amiga 500 con TV prescart	**L. 27.000**
Nuovo C64	Nuovo personal computer CPU 64K RAM computer ad alta risoluzione grafica, 256 combinazioni di colori, sintetizzatore di suono. Collegabile ad un qualsiasi televisore	**L. 375.000**
1764	Espansione di memoria per C64 256Kbytes	**L. 195.000**
C 128D	Personal computer CPU 128Kb RAM CPU 128Kbytes espandibile a 512Kbytes, 48Kbytes ROM, basic 7.0. Tastiera separata. Alta risoluzione grafica, 16 colori + 8 sprites. 40 80 (RGB) colonne. Programmabile in CP/M 3.0. Funzionante in modo C64. Floppy disk da 340Kb incorporato.	**L. 895.000**
1700	Espansione di memoria per C128. 128Kb	**L. 165.000**
1750	Espansione di memoria per C128. 512Kb	**L. 235.000**
	Registratori	
1530	Registratore compatibile C64 - C128 - C128D	**L. 55.000**
	Unità a dischi	
1541 II	Floppy disk drive 5" 1/4. Unità di memoria di massa, drive singolo, capacità 170Kbytes in linea. Compatibile con C64	**L. 395.000**
1581	Floppy disk drive 3" 1/2. Unità di memoria di massa, drive singolo da 3" 1/2, capacità 800Kbytes. Compatibile C64 - C128 - C128D	**L. 420.000**
A 1010	Floppy disk drive esterno 3" 1/2	**L. 395.000**
1802	Monitor a colori, alta risoluzione, 14" con audio. Collegabile al C64, C128, C128D (40 colonne)	
2080	Monitor a colori alta persistenza alta risoluzione, 14", con audio antiriflesso. Collegabile ad Amiga PC, C64, C128, C128D. 640x400 pixel, 4096 colori	**L. 630.000**
1084	Monitor a colori. Alta risoluzione, 14", con audio, antiriflesso. Collegabile al C64 - C128 - C128D - Amiga - PC. 640x400 pixel, 4096 colori.	**L. 545.000**
	Stampanti	
MPS 1200 S	Stampante. 80 colonne, 120 cps, bidirezionale, carta in modulo singolo e trascinamento modulo continuo per C64 - C128 - C128D	**L. 480.000**

MPS 1200 P	Stampante 80 colonne 120 cps, bidirezionale, carta in modulo singolo e trascinamento modulo continuo per A500	**L. 480.000**
MPS 1250	Stampante con stesse caratteristiche del modello MPS 1200 P, interfacce seriale RS232C e parallela Centronics di serie	**L. 495.000**
MPS 1500	Stampante a colori. 80 colonne, 130 cps, 4 colori, bidirezionale, carta in modulo singolo e trascinamento modulo continuo.	**L. 550.000**
	Accessori	
6499	Adattatore telematico omologato. Collegabile al C64, permette il collegamento a Videotel PGE-Banche Dati.	**L. 149.000**
1351	Mouse. Per C64, C128, C128D	**L. 99.000**
1311	Joystick. Comando per giochi	**L. 13.500**
1399	Joystick. 8 microswitch-autofire	**L. 29.000**

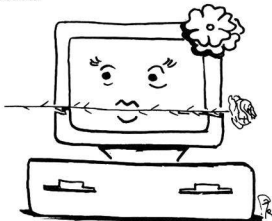
LISTINO PREZZI SISTEMI

Codice		**Prezzo (IVA esclusa)**
PCI	Microprocessore 8088 (coprocessore 8087 opzionale) frequenza clock 4,77MHz. RAM 512Kb. Espandibile a 640Kb; 1 floppy da 360Kb. Monitor a fosfori verdi 12".	**L. 945.000**
PC 10-III	Microprocessore Intel 8088 a 16 bit; clock 10 MHz; memoria RAM: 640 Kb; memoria ROM (BIOS): 8 Kb autoconfigurabile; sistema operativo MS-DOS 3.21; GW-Basic; scheda video integrata monocromatico/colore di serie con emulazioni CGA/MDA/Hercules via software; 2 floppy disk drive da 360 Kb; monitor monocromatico a fosfori verdi da 12"; porte seriale RS232C e parallela Centronics; tastiera avanzata 102 tasti con 12 tasti funzione; porta mouse, incluso controller per H.D.	**L. 1.490.000**
PC 10-IIIC	Stessa configurazione ma con monitor 14" a colori mod. 1084	**L. 1.890.000**
PC 20-III	Microprocessore Intel 8088 a 16 bit; memoria RAM da 640Kb; memoria ROM (BIOS) da 8Kb autoconfigurabile. Sistema operativo MS-DOS 3.21. Scheda video monocromatico/colore di serie con emulazioni CGA/MDA/Hercules via software. Un hard disk da 20Mb e un floppy disk drive da 360Kb. Monitor monocromatico a fosfori verdi da 12". Porta seriale RS232C parallela Centronics. Porta mouse incluso controller per H.D.; tastiera avanzata 102 tasti con 12 tasti funzione	**L. 2.240.000**
PC 20-IIIC	Stessa configurazione ma con monitor 14" a colori mod. 1084	**L. 2.640.000**
PC 40/20 AT	Microprocessore 80286 a 16 bit; clock 10/6 MHz; memoria RAM dal 1 Mb; hard disk da 20 Mb; un floppy disk drive da 1,2 Mb; Sistema operativo MS-DOS 3.2, GW-Basic; scheda video monocromatico/colore 132 colonne AGA di serie; monitor monocromatico a fosfori verdi da 14"; porta seriale RS232C e parallela Centronics; tastiera avanzata 102 tasti con 12 tasti funzione	**L. 3.490.000**
PC 40/40 AT	Microprocessore 80286 a 16 bit; clock 10/6 MHz; memoria RAM da 1Mb. Un hard disk da 40Mb; un floppy disk drive da 1,2Mb. Sistema operativo MS-DOS3.2, GW-Basic. Scheda video monocromatico/colore EGA WONDER di serie. Monitor monocromatico a fosfori verdi da 14". Porta seriale RS232 e parallela Centronics; tastiera avanzata 102 tasti con 12 tasti funzione	**L. 4.490.000**
PC 60/40	Microprocessore 80386 a 32 bit; clock 8/16 MHz; memoria RAM da 2,5 MB; hard disk da 40 Mb; 1 floppy disk drive da 1,2 Mb; sistema operativo MS-DOS 3.20; scheda EGA WONDER di serie; monitor ADI monocromatico a fosfori verdi da 14"; 2 porte seriali RS232C e 2 parallele Centronics. In opzione: floppy disk drive da 3" 1/2 e 1.44 Mb. Tastiera avanzata 102 tasti con 12 tasti funzione	**L. 8.490.000**
PC 60/80	Stessa configurazione del PC 60/40 ma con hard disk da 80 Mb e, in più, MS-DOS WINDOWS 386 e mouse	**L. 9.990.000**

Amiga 2000	Microprocessore Motorola 68000; clock 7,16 MHz; 1MB RAM; 256KB ROM; kickstart in ROM. Uscita PAL; 4 uscite a 4 voci su 2 canali (stereofonia). Amiga DOS; Amiga multitasking. 7 slots di sistema (2 combinati Amiga - Ibm PC AT); 1 slot video; 1 accesso diretto CPU 86 pin. Compatibilità MS-DOS interna con schede Janus (XT/AT CARD). Possibilità di un secondo drive interno da 3" 1/2 e di un drive interno da 5" 1/4. Monitor a colori 1084 tastiera; mouse. Workbench; Extras. MONITOR ESCLUSO	**L. 2.005.000**
	Monitor	
2080	Monitor a colori ad alta risoluzione e lunga persistenza 14" con audio; 640x400 pixel; 4096 colori per A500/A1000/A2000/PC AT	**L. 630.000**
1084	Monitor a colori ad alta risoluzione 14" con audio; antiriflesso; 640x400 pixel; 4096 colori per A500/A1000/A2000/PC AT	**L. 545.000**
1402	Monitor monocromatico da 12"	**L. 255.000**
1450	Monitor monocromatico BISYNC	**L. • □**
1950	Monitor colore BISYNC	**L. • □**
A 2024	Monitor monocromatico alta risoluzione a pagina bianca	**L. • □**
	Stampanti	
MPS 1200P	Stampante a 9 aghi 120 cps bidirezionale 80 colonne. Interfaccia Commodore per PC 10/III PC 40 PC/60/80	**L. 480.000**
MPS 1250	Stesse caratteristiche del modello MPS 1200P, interfacce seriale RS232C e parallela Centronics di serie	**L. 495.000**
MPS 1500	Stampante a colori a 9 aghi 130 cps; bidirezionale; 80 colonne; carta in modulo singolo e trascinamento modulo continuo; 4 colori; per Amiga e PC	
	Unità dischi	
K 910	Floppy disk drive aggiuntivo da 3" 1/2 e 1.44 Mb per PC 10-III e PC 20-III	**L. 295.000**
A 1010	Floppy disk drive esterno da 3" 1/2	**L. 395.000**
A 2010	Floppy disk drive interno da 3" 1/2	**L. 290.000**
A 2092 + PC 5060	Hard disk da 20 Mb settorizzabile PC/Amiga + scheda controller per hard disk per A 2000	**L. 845.000**
A 2080 + A 2020	Scheda Janus XT compatibile + drive	**L. 895.000**
A 2286 + A 2020	Scheda Janus II AT compatibile + drive	**L. 1.285.000**
A 2620	Scheda 68020 con processore a 32 bit; 14,28 MHZ	**L. •**
A 2090 + A 2092	Hard disk 20 Mb; interfaccia SCSI per A 2000	**L. 1.025.000**
A 2090 + A 2094	Hard disk 40 Mb; interfaccia SCSI per A 2000	**L. 1.575.000**
A 590	Hard disk da 20 Mb + fast RAM per A 500	**L. • □**
	Accessori	
PC EXP1	Box esterno per espansione PCI; può contenere 3 schede aggiuntive e hard disk alimentazione e cavi compresi	**L. 590.000**
A 2052	Scheda di espansione RAM da 2 Mb per A 2000 (max 4)	**L. 590.000**
A 2060	Scheda video/RF molulatore per TV e VCR per A2000	**L. 138.000**
A 2058	Espansione di memoria da 8 Mb RAM per A2000	**L. 3.760.000**
A UNIX	Scheda UNIX utilizzabile con A2620 (68020 board)	**L. • □**
1352	Mouse due tasti collegabile alla serie dei PC include Microsoft Mouse Driver	**L. 72.000**
A 2300	Genlock card semiprofessionale per A2000; permette di miscelare grafici con videocamere VCR; selettore per video Amiga, monitors esterni, proiettori; include software per titolazione	**L. 345.000 Δ**
A 2350	Professional Video Adapter Card per A 2000. Genlock professionale con freeze frame e digitalizzatore video incluse software di controllo per la gestione interattiva	**L. 1.645.000 Δ**

• prezzo da definire
Δ disponibile da giugno '88
□ disponibile da settembre '88

I COMMODORE POINT

I primi 100 negozi selezionati dalla "Commodore" per la vendita dei suoi prodotti

Al Risparmio	*V.le Monza 204*	20128 Milano	Tel. 02-2573440
Braha Alberto S.D.F.	*Via Pier Capponi 5*	20145 Milano	Tel. 02-437468
E.D.S.	*C.so Porta Ticinese 4*	20123 Milano	Tel. 02-8322045
ESC	*Via Roggia Scagna 7*	20127 Milano	Tel. 02-2871300
Farel	*Via A. Volta 21*	20121 Milano	Tel. 02-650042
Giglioni	*Via D'Ovidio 8*	20131 Milano	Tel. 02-2360397
Giglioni	*V.le Luigi Sturzo 45*	20154 Milano	Tel. 02-654906
Logitek	*Via Golgi 60*	20133 Milano	Tel. 02-538931
Marcucci	*Via F.lli Bronzetti 37*	20129 Milano	Tel. 02-7386051
Melchioni	*Via P. Colletta 37*	20135 Milano	Tel. 02-57941
Messaggerie Musicali	*Galleria Del Corso*	20100 Milano	Tel. 02-50841
Newel	*Via Mac Mahon 75*	20155 Milano	Tel. 02-323492
Rivola	*Via Vitruvio 43*	20124 Milano	Tel. 02-6694160
F.lli Galimberti	*Via Naz. dei Giovi 28/36*	20030 Barlassina (Mi)	Tel. 0362-560625
P. Giorgio Ostellari	*Via Molino Arese 65*	20031 Cesano Mad.(Mi)	Tel. 0362-504392
P. Giorgio Ostellari	*Via Milano 300*	20233 Desio (Mi)	Tel. 0362-621042
GBC Italiana	*V.le Matteotti 66*	20092 Cinisello B. (Mi)	Tel. 02-6181801/ 6189391
	Via Petrella 6	20100 Milano	Tel. 02-203607/8
	Via Cantoni 7	20100 Milano	Tel. 02-437478
Casa Della Musica	*Via Indipendenza 21*	20093 Cologno Mon.(Mi)	Tel. 02-2542117
Penati	*Via Verdi 28/30*	20011 Corbetta (Mi)	Tel. 02-9779401
EPM System	*V.le Italia 12*	20094 Corsico (Mi)	Tel. 02-4407979
Centro Comp. Pandolfi	*Via Corridoni 18*	20025 Legnano (Mi)	Tel. 0331-546426
Computeam	*Via Vecellio 41*	20035 Lissone (Mi)	Tel. 039-481010
Futura	*Via Solferino 31*	20075 Lodi (Mi)	Tel. 0371-54457
MBM Inf. Sys.	*Corso Roma 112*	20075 Lodi (Mi)	Tel. 0371-53610
L'Amico del computer	*V.le Lombardia 17*	20077 Melegnano (Mi)	Tel. 02-9838341
Bit 84	*Via Italia 4*	20052 Monza (Mi)	Tel. 039-320813
I.C.O.	*Via dei Tigli 14*	20090 Opera (Mi)	Tel. 02-5242146
Comif	*Via Autolinee 10*	24100 Bergamo	Tel. 035-218553
Cordani	*Via dei Caniana 8*	24100 Bergamo	Tel. 035-258184
D.R.B.	*Via Borgo Palazzo 65*	24100 Bergamo	Tel. 035-237292
New Systems	*Via Paglia 36*	24100 Bergamo	Tel. 035-248109
Computer Team hi-tec	*Via Verdi 1/B*	24030 Carvico (Bg)	Tel. 035-790244
Ott. Optometr. Rovetta	*P.zza Garibaldi 6*	24065 Lovere (Bg)	Tel. 035-960705
A.I.S. International	*Via San Carlo 25*	24016 S.Pellegrino T.(Bg)	Tel. 0345-22662
Sisthema	*Via Roma 45*	24067 Sarnico (Bg)	Tel. 035-910750
Computer Center	*Via Cipro 62*	25125 Brescia	Tel. 030-223230
Informatica 2000	*Via Stazione 16/B*	25100 Brescia	Tel. 030-54015
Vigasio Mario	*Portici Zanardelli 3*	25100 Brescia	Tel. 030-59330/ 295858
Mister Bit	*Via Mazzini 70*	25043 Breno (Bs)	Tel. 0364-22835
Cavalli Pietro	*Via 10 Giornate 14B*	25030 Castrezzato (Bs)	Tel. 030-714013
Vietti Giuseppe	*Via Milano 1/B*	25032 Chiari (BS)	Tel. 030-7100206
Megabyte	*P.zza Duomo 17*	25015 Desenzano G.(Bs)	Tel. 030-9144880
Ditta Resi Rino	*Via XX Settembre 7*	25016 Ghedi (BS)	Tel. 030-901224
Info Cam	*Provinciale 38*	25050 Gratacasolo (Bs)	Tel. 0364-89379
Il Computer	*Via Indipendenza 90*	22100 Como	Tel. 031-240959
2M Elettronica	*Via Sacco 3*	22100 Como	Tel. 031-278227
Eletrongros	*Via L. Da Vinci 54*	22062 Barzanò (Co)	Tel. 039-957318
Ega	*Via Mazzini 42*	22065 Cassago Br.(Co)	Tel. 039-956307
Ega	*Via Aldo Moro 17*	22043 Galbiate (Co)	Tel. 0341-522028
Data Found Comp. Sh.	*Via A. Volta 4*	22036 Erba (Co)	Tel. 031-645761
Righi Elettronica	*Via G. Leopardi 26*	22077 Olgiate Com. (Co)	Tel. 031-946766
Fumagalli	*Via Cairoli 48*	22053 Lecco (Co)	Tel. 0341-863341
Cima Eelettronica	*Via Leonardo Da Vinci 7*	22053 Lecco (Co)	Tel. 0341-371106
Mondo Computer	*Via Giuseppina 11/B*	26100 Cremona	Tel. 0372-882079
Prisma	*Via Buoso Da Dovara 8*	26100 Cremona	Tel. 0372-437900
Telco	*P.zza Marconi 2/A*	26100 Cremona	Tel. 0372-31544
Elcom/GBC	*Via IV Novembre 56/58*	26013 Crema (Cr)	Tel. 0373-83393
Euroelettronica	*Via XX Settembre*	26013 Crema (Cr)	Tel. 0373-86966
Kofbaker	*Via Marchi 65/B*	26039 Vescovato (CR)	Tel. 0372-818601
Computer	*Galleria Fermi 7*	46100 Mantova	Tel. 0376-325616
32 Bit (Comp. St.)	*Via Cesare Battisti 14*	46100 Mantova	Tel. 0376-326770
Elettronica di Basso	*V.le Risorgimento 69*	46100 Mantova	Tel. 0376-329311
Poliware	*C.so C. Alberto 76*	27100 Pavia	Tel. 0382-22636
Log. Infor. Com. Shop	*V.le Monte Grappa 32*	27029 Vigevano (Pv)	Tel. 0381-81883
M. Visentin	*C.so Vitt. Emanuele 76*	27029 Vigevano (Pv)	Tel. 0381-83833
Computer Line	*Via G. Carducci 4*	29100 Piacenza	Tel. 0523-30691
Delta Computer	*Via Mart. della Resistenza 15/4*	29100 Piacenza	Tel. 0523-753318
Sover	*Via IV Novembre 60*	29100 Piacenza	Tel. 0523-34388
Cipolla Mauro	*Via Tremogge 25*	23100 Sondrio	Tel. 0342-213569
Fotonova		23010 S. Pietro Ber. (So)	Tel. 0342-492319
Dimeco Sistemi	*Via Garibaldi*	21100 Varese	Tel. 0332-237201
Il Centro Elettronico	*Via Morazzone 2*	21100 Varese	Tel. 0332-231006
Supergames	*Via Carrobbio 13*	21100 Varese	Tel. 0332-241092
Busto Bit	*Via Gavinana 17*	21052 Busto Arsizio (Va)	Tel. 0331-625034
Crespi Giuseppe & C.	*V.le Lombardia 59*	21053 Castellanza (Va)	Tel. 0331-503023
Computer Shop	*Via A. Da Brescia 2*	21013 Gallarate (Va)	Tel. 0331-798612
Lima Import-Export	*Via Clerici 196*	21040 Gerenzano (Va)	Tel.
J.A.C. nuove tecn.	*Via Matteotti 38*	21018 Sesto Cal. (Va)	Tel. 0331-923134
Bit Micro	*Via Mazzini 102*	15100 Alessandria	Tel. 0131-443252
West Records	*C.so Roma 85*	15100 Alessandria	Tel. 0131-441090
S.G.E. Elettronica	*Via Bandello 19*	15057 Tortona (Al)	Tel. 0131-867709
Record di Forina G.	*C.so Alfieri 166/3*	14100 Asti	Tel. 0141-34240
Il Diagramma Italia	*P.le Libertà 4*	12100 Cuneo	Tel. 0171-57067
Rossi Computers	*C.so Nizza 42*	12100 Cuneo	Tel. 0171-2339
Punto Bit	*C.so Langhe 26/C*	12051 Alba (Cn)	Tel. 0173-49833
SDI	*Via Vitt. Emanuele 250*	12042 BRA (CN)	Tel. 0172-421392
Aschieri Gianfranco	*C.so Eman. Filiberto 6*	12045 Fossano (CN)	Tel. 0172-62995
Curetti Augusto	*C.so Italia 3*	12054 Mondovì (Cn)	Tel. 0174-42014
Ditta Elettrogamma	*C.so Risorgimento 20*	28100 Novara	Tel. 0321-176358
Elcom	*C.so Mazzini 11*	28100 Novara	Tel. 0321-391293
Programma 3	*V.le Buonarroti 8*	28100 Novara	Tel. 0321-36367/ 399903
Punto Video	*C.so Risorgimento 391*	28100 Novara	Tel. 0321-477367
Computer	*Via Monte Zeda 4*	28041 Arona (No)	Tel.
All Computer	*C.so Garibaldi 106*	28021 Borgomanero (No)	Tel. 0322-844142
Micrologic	*Via Giovanni XIII 2*	28037 Domodossola (No)	Tel.
Elliott Computer Shop	*Via Don Minzoni 32*	28044 Intra (No)	Tel. 0323-43517
ABA Elettronica	*Via C. Fossati 5/P*	10100 Torino	Tel. 011-302065
Alex Computer e giochi	*C.so Francia 333/4*	10142 Torino	Tel. 011-7730184
C.D.M. Elettronica	*Via Marochetti 17*	10126 Torino	Tel. 011-636345/ 634900
Computing New	*Via Marco Polo 40/E*	10129 Torino	Tel. 011-501512
De Bug	*C.so Vitt. Emanuele II 22*	10100 Torino	Tel. 011-832986
Desme Universal	*Via San Secondo 95*	10100 Torino	Tel. 011-592551
F.D.S.	*Via Borgaro 86/D*	10100 Torino	Tel. 011-2168900
Computer Home	*Via San Donato 46/D*	10100 Torino	Tel. 011-4731196
Informatica Italia	*C.so Re Umberto 129*	10128 Torino	Tel. 011-501647
MT Informatica	*C.so Giulio Cesare 58*	10100 Torino	Tel. 011-850955/ 238803
New Bus. Computer	*Via Nizza 45*	10100 Torino	Tel. 011-214235
Radio Tv Mirafiori	*C.so Un. Sovietica 381*	10135 Torino	Tel. 011-616190/ 6197189
SMT Elettronica	*Via Bibiana 83/B*	10147 Torino	Tel. 011-218243
Paul e Chico Videos.	*Via Vitt. Emanuele 52*	10023 Chieri (To)	Tel. 011-9470295
Bit Informatica	*Via V. Emanuele 154*	10073 Ciriè (To)	Tel. 011-9205455
HI-FI Club	*C.so Francia 92C*	10093 Collegno (To)	Tel. 011-4110256
ICS	*Stradale Torino 73*	10015 Ivrea (To)	Tel. 0125-251797
Cerrutti Mauro	*C.so Torino 234*	10064 Pinerolo (To)	Tel. 0121-70316
Eurex	*C.so Indipendenza 5*	10086 Rivarolo C.se (To)	Tel. 0124-27984
Ditta Elettrogamma	*C.so Bormida*	13100 Vercelli	Tel. 0161-53689
Elettronica	*Strada Torino 15*	13100 Vercelli	Tel. 0161-393163
C.S.I. Teorema	*Via Losana 9*	13051 Biella (Vc)	Tel. 015-28622
Fotostudio Trevisan	*Via XXV Aprile 24/B*	13014 Cossato (VC)	Tel. 015-921431
Studio Fotogr. Imarisio	*P.zza Martiri Libertà 7*	13039 Trino (Vc)	Tek 0161-82081
F.lli Gatti	*Via Festaz 75*	11100 Aosta	Tel. 0165-35659

PRODOTTI SYSTEMS EDITORIALE

Software su cassetta

La voce III	L.12000
Raffaello	L.10000
Oroscopo	L.12000
Computer-Music	L.12000
Gestione familiare	L.12000
Banca dati	L.12000
Dichiarazione dei redditi (740/S)	L.16000
Matematica finanziaria	L.20000
Analisi di bilancio	L.20000
Arredare (richiede linguaggio Simon's Basic)	L.10000

Software su disco

Ms-Dos & Gw-Basic	L.25000
Ms-Dos & Gw-Basic (con prova di acquisto cassetta)	L.15000
La voce III	L.12000
Raffaello	L.10000
Oroscopo	L.12000
Computer-Music	L.12000
Gestione familiare	L.12000
Banca dati	L.12000
Dichiarazione dei redditi (740/S)	L.24000
Matematica finanziaria	L.20000
Analisi di bilancio	L.20000
Arredare (richiede linguaggio Simon's Basic)	L.20000
Graphic Expander C/128 in modo 80 colonne	L.27000
Linguaggio macchina + Routine grafiche	L.12000

Offerta "Commodore speciale L.M." + dischetto L.16000

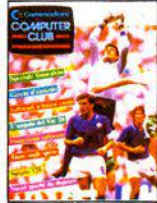
Libri

64 programmi per il Commodore 64	L. 4800
I miei amici C/16 e Plus/4	L. 7000
Strategie vincenti per Commodore 64	L. 5800
62 programmi per Vic 20, C/16 e Plus/4	L. 6500
Utilities e giochi didattici	L. 6500
Tutti i segreti dello Spectrum	L. 7000
Simulazioni e test per la didattica	L. 7000
Impara giocando il Basic dello Spectrum	L. 7000
Micro Pascal per Commodore 64/128	L. 7000
Dal registratore al drive del C/64	L. 7000
Ada	L. 5000
Il linguaggio Pascal	L. 5000

Directory

Ciascun dischetto	L.12000

Arretrati

Ciascun numero arretrato di Commodore Computer Club	L. 5000
Ciascun numero arretrato di Personal Computer	L. 5000
Ciascun numero arretrato di VR Videoregistrare	L. 5000

Per un ottimale utilizzo del software "Matematica finanziaria" è opportuna la lettura degli articoli relativi pubblicati sui N.13, 14, 15 della rivista "Commodore" e sui N.1, 2 e 3 della Rivista Personal Computer.

Per un ottimale utilizzo del software "Analisi di Bilancio" è opportuna la lettura degli articoli relativi pubblicati sui N.2, 3, 5 della Rivista Personal Computer.

Per un ottimale utilizzo del software "Linguaggio Macchina e Routine grafiche per C/64" è opportuna la lettura del fascicolo "Commodore Speciale" appositamente dedicato.

Coloro che desiderano procurarsi i prodotti della Systems Editoriale devono inviare, oltre alla cifra risultante dalla somma dei singoli prodotti, la cifra di L.3000 per spese di imballo e spedizione, oppure L.6000 se si preferisce la spedizione per mezzo raccomandata.

Sconti e agevolazioni

Le spese di imballo e spedizione sono a carico della Systems se ciascun ordine è pari ad almeno L.50000 (di listino).

Gli abbonati hanno diritto allo sconto del 10% e alla spedizione gratuita se la somma totale raggiunge la cifra di L.50000 (di listino).

Oltre alla spedizione gratuita, viene praticato uno sconto del 10% (per gli abbonati è del 20%) se la cifra raggiunta per ciascun ordine raggiunge le L.100000 (di listino).

Abbonamenti

Commodore Computer Club (11 fascicoli)	L.45000
Personal Computer (11 fascicoli)	L.40000
Commodore Computer Club + Personal Computer (11 + 11 fascicoli)	L.70000
VR Videoregistrare (12 numeri)	L.45000

N.B.: la cifra per gli abbonamenti non può essere conteggiata per ottenere gli sconti e le agevolazioni di cui sopra.

Non è assolutamente possibile inviare materiale contrassegno.

Compilate un normale modulo di C/C postale indirizzando a:

C/C postale N. 37952207
Systems Editoriale
Viale Famagosta, 75
20142 Milano

Non dimenticate di indicare chiaramente, sul retro del modulo (nello spazio indicato con "Causale del versamento") non solo il vostro nominativo completo di recapito telefonico, ma anche il materiale desiderato.

In ogni caso sarebbe opportuno inviare la presente scheda, debitamente compilata, allegando la fotocopia della ricevuta del versamento effettuato.

Chi volesse ricevere più celermente la confezione deve inviare la somma richiesta mediante assegno circolare oppure normale assegno bancario (non trasferibile o barrato due volte) intestato a: Systems Editoriale - Milano.

IN EDICOLA

N. 5 - L. 12.000

Commodore 64 Club

- Cover
- Terrore a Dunwich
- Hacker
- Lister
- Amadeus' Revenge
- Midnight
- Hawk mission
- Dispenser

4 GAMES
2 UTILITY
3 ADVENTURE

Dischetto a 2 facce, oltre 300 Kbyte di software

Commodore Club - Dir. Resp. A. Ronchetti Edizioni Systems Editoriale Srl - V.le Famagosta 75 -20142 Milano - Reg. Trib. MI. n. 104 del 25/2/84 - Distr. MePe.

IN EDICOLA

Lire 15.000
Commodore 64 Club
100% Turbo
100% Originale
Serie Dylan Dog # 1
DYLAN DOG
• LE NOTTI DELLA LUNA PIENA
• IL CASTELLO DELLE ILLUSIONI
Systems
LE NOTTI DELLA LUNA PIENA